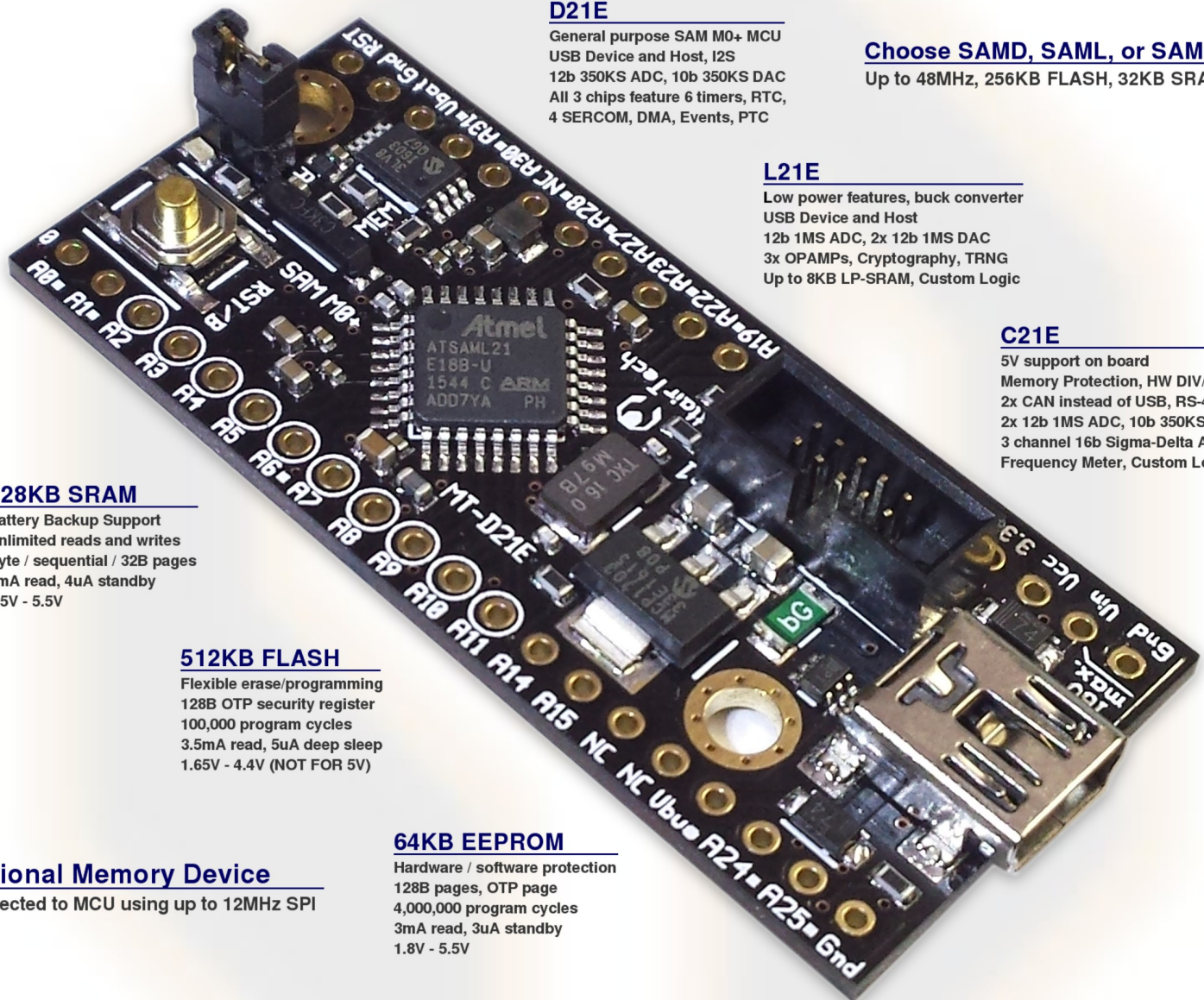


Microchip® / Atmel® ARM® Cortex® M0+ Development Board



D21E

General purpose SAM M0+ MCU
USB Device and Host, I2S
12b 350KS ADC, 10b 350KS DAC
All 3 chips feature 6 timers, RTC,
4 SERCOM, DMA, Events, PTC

Choose SAMD, SAML, or SAMC

Up to 48MHz, 256KB FLASH, 32KB SRAM

L21E

Low power features, buck converter
USB Device and Host
12b 1MS ADC, 2x 12b 1MS DAC
3x OPAMPs, Cryptography, TRNG
Up to 8KB LP-SRAM, Custom Logic

C21E

5V support on board
Memory Protection, HW DIV/SQRT
2x CAN instead of USB, RS-485
2x 12b 1MS ADC, 10b 350KS DAC
3 channel 16b Sigma-Delta ADC
Frequency Meter, Custom Logic

128KB SRAM

Battery Backup Support
Unlimited reads and writes
Byte / sequential / 32B pages
3mA read, 4uA standby
2.5V - 5.5V

512KB FLASH

Flexible erase/programming
128B OTP security register
100,000 program cycles
3.5mA read, 5uA deep sleep
1.65V - 4.4V (NOT FOR 5V)

64KB EEPROM

Hardware / software protection
128B pages, OTP page
4,000,000 program cycles
3mA read, 3uA standby
1.8V - 5.5V

Optional Memory Device

Connected to MCU using up to 12MHz SPI

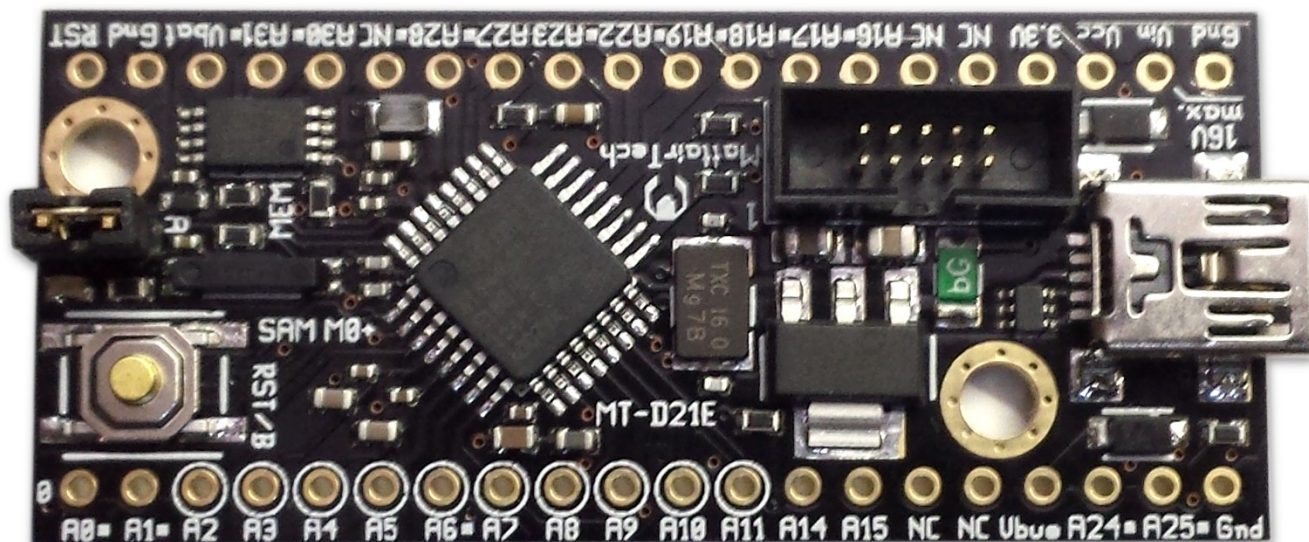
Revision B

Table of Contents

Overview.....	4
Introduction.....	4
Board Features.....	5
Configuration Options.....	6
Board Revisions.....	6
Microcontroller Features / Comparisons.....	7
Development Software Matrix.....	9
Pinout.....	10
Solder Jumpers.....	14
Memory Device.....	17
128KB SRAM with Battery Backup.....	17
512KB FLASH.....	17
64KB EEPROM.....	18
Features (from ON Semiconductor datasheet).....	18
Power Supply.....	19
Other Hardware.....	20
Schematic.....	21
Parts List.....	22
Dimension Drawing.....	24
MattairTech Arduino SAM M0+ Core.....	25
What's New Beta (1.6.8-beta).....	25
What's New Release (1.6.6).....	26
Features Summary.....	27
Board Variants.....	28
Pin Configurations.....	29
Tools Menu Additions.....	30
Microcontroller Menu.....	30
Clock Source Menu.....	30
Bootloader Size Menu.....	30
Serial Config Menu.....	30
USB Config Menu.....	30
Clock Source.....	31
External 32.768KHz Crystal.....	31
External High-Speed Crystal.....	31
Internal Oscillator.....	31
Internal Oscillator with USB Calibration.....	31
Clock Generators.....	32
Analog Reference.....	32
Chip Specific Notes.....	33
SAMD21.....	33

SAML21.....	33
SAMC21.....	33
SAMD11.....	34
Reducing SRAM/FLASH Usage on the D11.....	34
Differences Between MattairTech and Arduino Cores.....	34
Random Notes (TODO).....	34
Serial Monitor.....	35
Installation.....	37
Driver Installation.....	37
Windows.....	37
Linux.....	37
OS X.....	38
SAM M0+ Core Installation.....	38
Uploading the First Sketch.....	39
Beta Builds.....	39
New PinDescription Table.....	39
Possible Future Additions/Changes.....	40
ChangeLog.....	41
Troubleshooting.....	41
SAM-BA USB CDC Bootloader (Arduino Compatible).....	43
Bootloader Firmware Installation.....	44
Bootloader Installation Using the Arduino IDE.....	44
Bootloader Installation Using Another Tool (ie: Atmel Studio, openocd).....	44
Bootloader Binaries.....	44
MattairTech Boards.....	44
Arduino/Genuino Boards.....	45
Generic Boards.....	45
Using Bossac Standalone.....	45
USB Mass Storage Bootloader.....	47
Blink Demo.....	49
Troubleshooting / FAQ.....	50
Support Information.....	50
Legal.....	51
Appendix A: Precautions.....	56
Appendix B: Other MattairTech Products.....	57

Overview



Introduction

The MattairTech MT-D21E is a development board for the 32-pin Microchip / Atmel SAMx21E ARM Cortex M0+ microcontrollers. Choose between the D21E (general purpose M0+ MCU, USB, I2S, also used in the Arduino Zero), L21E (low power features, enhanced analog, USB, crypto/TRNG, custom logic), or C21E (5V support, MPU, 2x CAN instead of USB, Sigma-Delta and 2x SAR ADCs). An optional SPI serial memory device can be mounted (128KB SRAM w/backup, 512KB FLASH, or 64KB EEPROM). The board can be powered from USB or the Vin pin, with 2 schottky diodes for simple switching and reverse-polarity protection. This voltage is regulated to 3.3V by the onboard 250mA, extremely low quiescent current (2uA) LDO regulator which supports up to 16V on Vin. The C21 supports 5V operation by connecting USB Vbus to Vcc (jumpers). A 250mA (hold) PTC resettable fuse provides overcurrent protection. Also mounted is a mini USB connector with ESD protection, blue LED, 16MHz crystal, 32.768KHz crystal, button, jumper, and Vbat input. Arduino 1.6.x compatible core files (1.6.x, 1.8.x IDE) for all 3 chips are provided. A SAM-BA USB CDC bootloader (TTL serial only for C21) is preinstalled for programming without an external tool (Arduino or standalone use). The Cortex header (10-pin, 50-mil) supports an external debugger / programmer. There are 40 dual inline header pins with 0.1" pin spacing and 0.7" row spacing which supports breadboard or perfboard mounting. The PCB measures 2.1" x 0.9" x 0.062" (52mm x 23mm x 1.6mm) with two 3mm mounting holes.

Board Features

- **Microchip / Atmel ARM Cortex M0+ 32-pin microcontroller**
 - **Choice of SAMD, SAML, or SAMC chips**
 - **D21E** (general purpose SAM M0+ MCU, USB, I2S, also used in the Arduino Zero)
 - **L21E** (low power features, enhanced analog, USB, crypto/TRNG, custom logic)
 - **C21E** (5V support, MPU, 2x CAN instead of USB, Sigma-Delta and 2x SAR ADCs)
 - CPU operates at up to 48MHz
 - All chips available with up to 256KB FLASH and 32KB of SRAM
 - All chips feature 6 timers, RTC, 4 SERCOM, DMA, PTC, AC
- **Optional SPI Serial Memory Device**
 - 128KB SRAM with battery backup (Vbat pin)
 - 512KB FLASH with protection features and OTP page
 - 64KB EEPROM with OTP page
- **Onboard 3.3V, 250mA LDO regulator**
 - Supports up to 16V DC input on Vin pin
 - Extremely low quiescent current (2.0uA typical)
 - Over-current and over-temperature protection
- **Powered by USB Vbus or external power source on Vin (up to 16V)**
 - Simple diode power source switching, reverse-polarity protection
 - With the C21 installed, USB Vbus can be connected to Vcc for 5V operation
- **Arduino 1.6.x compatible core (1.6.x, 1.8.x IDE)**
- **SAM-BA USB CDC bootloader (D21 and L21 only, C21 has serial bootloader)**
 - Arduino compatible (use the Arduino IDE to upload)
- Bossa command line utility (Windows, Linux, limited OS X)
- **PTC resettable fuse (250mA hold / 500mA trip)**
- **Cortex Debug Header (10-pin, 50-mil) for device programming and debugging**
- **16MHz crystal (use PLL for up to 48MHz cpu clock)**
- **32.768KHz crystal (use PLL or FLL for up to 48MHz cpu clock)**
- **Blue Status LED (can be disconnected)**
- Jumper A for general use (pin A27)
- **Button B configurable for reset or general use (pin A31) with debouncing**
- **Two 4.7Kohm resistors can be connected to pins A16 and A17 for use with I2C**
- **Mini USB connector with ESD protection on USB Vbus, D+, and D- lines**
- USB pins (or CAN on the C21) routed to header pins (for panel-mount connector)
- Ferrite bead and 2 capacitors on analog supply
- Two capacitors each can be enabled for pins A3 and/or A4 for use with external references
- **24 solder jumpers on PCB bottom for configuration flexibility**
- All PORT pins can be routed to headers
- 2 main headers are on 0.1" spacing (breadboard/perfboard mounting), 0.7" apart
- **Two 3mm mounting holes (~5mm pad)**
- High-quality PCB with gold-plated finish
- Measures approx. 2.1" x 0.9" (52mm x 23mm) and 0.062" (1.6mm) thick

Configuration Options

Choose SAMD, SAML, or SAMC (usually only 256KB version, see docs for more MCU info):

- **D21E** (general purpose SAM M0+ MCU, USB, I2S, also used in the Arduino Zero)
- **L21E** (low power features, enhanced analog peripherals, USB, crypto/TRNG, custom logic)
 - Inductor installed (in place of 0ohm resistor) to support on-chip buck converter, no pin A28
- **C21E** (5V support on board, MPU, 2x CAN instead of USB, Sigma-Delta and 2x SAR ADCs)
 - Can connect Vcc (default 3.3V) to 5V from USB Vbus. DO NOT use FLASH with 5V.

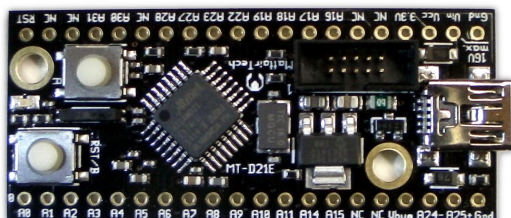
Choose Optional Memory Device (will connect to MCU using SPI at up to 12MHz):

- **128KB SRAM** with battery backup
 - Can connect Vbat header pin (coin cell) to SRAM backup pin (Vbat)
- **512KB FLASH** with protection features and OTP page
- **64KB EEPROM** with OTP page

All come with SAM-BA bootloader preinstalled:

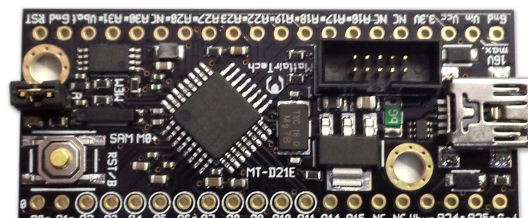
- USB CDC Serial supported with D21 and L21
- TTL Serial supported with C21

Board Revisions



Revision A (until March 14, 2017)

[MT-D21E User Guide.pdf](#)



Revision B (released March 15, 2017)

[MT-D21E revB User Guide.pdf](#) (this manual)

Revision B includes the following changes:

- Add support for more MCU options (SAMD, SAML, or SAMC)
 - With the SAML, an inductor is installed to support the buck converter
 - 5V from USB Vbus can be connected to Vcc to support SAMC at 5V (through 2 jumpers)
- Serial memory device can be installed (128K SRAM, 512KB FLASH, or 64KB EEPROM)
 - Using SPI at up to 12MHz using pins A18, A19, A22 (A27 for CS)
- Added Vbat and Gnd pins (coin cell), can connect to SRAM Vbat and/or Vcc (advanced)
- Button A now jumper A to make room for memory device, used as CS for memory as well
- Moved the blue LED to pin A6 (arduino pin 6), changed resistor to 1Kohm (600uA @ 3.3V)
- Physically larger PTC fuse (0805) with higher trip (500mA) and hold (250mA)
- Changed ESD device to protect Vbus in addition to D+ and D-

Microcontroller Features / Comparisons

	ATSAMD21E	ATSAML21E	ATSAMC21E
Product Presentation	http://atmel.force.com/support/servlet/fileField?id=0BEG000000003jc	http://atmel.force.com/support/servlet/fileField?id=0BEG000000002Wx	http://atmel.force.com/support/servlet/fileField?id=0BEG000000003Hq
Datasheet	http://ww1.microchip.com/downloads/en/DeviceDoc/40001882A.pdf	http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-42385-SAM-L21-Datasheet.pdf	http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-42365-SAM-C21_Datasheet.pdf
Voltage	● 1.62V - 3.63V	● 1.62V - 3.63V	● 2.7V - 5.5V
Processor	<ul style="list-style-type: none"> ● ARM Cortex-M0+ CPU running at up to 48MHz ● Single-cycle hardware multiplier ● Micro Trace Buffer 	<ul style="list-style-type: none"> ● ARM Cortex-M0+ CPU running at up to 48MHz ● Single-cycle hardware multiplier ● Micro Trace Buffer 	<ul style="list-style-type: none"> ● ARM Cortex-M0+ CPU running at up to 48MHz ● Single-cycle hardware multiplier ● Micro Trace Buffer ● Memory Protection Unit (MPU)
Memories	<ul style="list-style-type: none"> ● 32/64/128/256KB in-system self-programmable Flash ● 4/8/16/32KB SRAM Memory 	<ul style="list-style-type: none"> ● 32/64/128/256KB in-system self-programmable Flash ● 1/2/4/8KB Flash Read-While-Write section ● 4/8/16/32KB SRAM Memory ● 2/4/8KB SRAM Low power Memory 	<ul style="list-style-type: none"> ● 32/64/128/256KB in-system self-programmable Flash ● 1/2/4/8KB independent self-programmable Flash for EEPROM emulation ● 4/8/16/32KB SRAM Memory
System	<ul style="list-style-type: none"> ● Power-on reset (POR) and brown-out detection (BOD) ● Internal and external clock options with 48MHz Digital Frequency Locked Loop (DFLL48M) and 48MHz to 96MHz Fractional Digital Phase Locked Loop (FDPLL96M) ● External Interrupt Controller (EIC) ● 16 external interrupts ● One non-maskable interrupt ● Two-pin Serial Wire Debug (SWD) programming, test and debugging interface 	<ul style="list-style-type: none"> ● Power-on reset (POR) and brown-out detection (BOD) ● Internal and external clock options with 48MHz Digital Frequency Locked Loop (DFLL48M) and 48MHz to 96MHz Fractional Digital Phase Locked Loop (FDPLL96M) ● External Interrupt Controller (EIC) ● 16 external interrupts ● One non-maskable interrupt ● Two-pin Serial Wire Debug (SWD) programming, test and debugging interface 	<ul style="list-style-type: none"> ● Power-on reset (POR) and brown-out detection (BOD) ● Internal and external clock options with 48MHz to 96MHz Fractional Digital Phase Locked Loop (FDPLL96M) ● External Interrupt Controller (EIC) ● 16 external interrupts ● One non-maskable interrupt ● Two-pin Serial Wire Debug (SWD) programming, test and debugging interface
Low Power	<ul style="list-style-type: none"> ● Idle and standby sleep modes ● SleepWalking peripherals 	<ul style="list-style-type: none"> ● Idle, Standby, Backup, and Off sleep modes ● SleepWalking peripherals ● Static and Dynamic Power Gating Architecture ● Battery backup support (NOT E) ● Two Performance Levels ● Embedded Buck/LDO regulator supporting on-the-fly selection 	<ul style="list-style-type: none"> ● Idle, standby, and off sleep modes ● SleepWalking peripherals
DMA	<ul style="list-style-type: none"> ● 12-channel Direct Memory Access Controller (DMAC) 	<ul style="list-style-type: none"> ● 16-channel Direct Memory Access Controller (DMAC) 	<ul style="list-style-type: none"> ● 12-channel Direct Memory Access Controller (DMAC)
Event System	<ul style="list-style-type: none"> ● 12-channel Event System 	<ul style="list-style-type: none"> ● 12-channel Event System 	<ul style="list-style-type: none"> ● 12-channel Event System

16-bit Timers	<ul style="list-style-type: none"> Three 16-bit Timer/Counters (TC), configurable as either: <ul style="list-style-type: none"> One 16-bit TC with compare/capture channels One 8-bit TC with compare/capture channels One 32-bit TC with compare/capture channels, by using two TCs 	<ul style="list-style-type: none"> Three 16-bit Timer/Counters (TC) including one low-power TC, each configurable as: <ul style="list-style-type: none"> One 16-bit TC with two compare/capture channels One 8-bit TC with two compare/capture channels One 32-bit TC with two compare/capture channels, by using two TCs 	<ul style="list-style-type: none"> Three 16-bit Timer/Counters (TC), configurable as either: <ul style="list-style-type: none"> One 16-bit TC with compare/capture channels One 8-bit TC with compare/capture channels One 32-bit TC with compare/capture channels, by using two TCs
24-bit Timers	<ul style="list-style-type: none"> Two 24-bit Timer/Counters and one 16-bit Timer/Counter for Control (TCC), with extended functions: <ul style="list-style-type: none"> Up to four compare channels with optional complementary output Generation of synchronized pulse width modulation (PWM) pattern across port pins Deterministic fault protection, fast decay and configurable dead-time between complementary outputs Dithering that increase resolution with up to 5 bit and reduce quantization error 	<ul style="list-style-type: none"> Two 24-bit Timer/Counters and one 16-bit Timer/Counter for Control (TCC), with extended functions: <ul style="list-style-type: none"> Up to four compare channels with optional complementary output Generation of synchronized pulse width modulation (PWM) pattern across port pins Deterministic fault protection, fast decay and configurable dead-time between complementary outputs Dithering that increase resolution with up to 5 bit and reduce quantization error 	<ul style="list-style-type: none"> Two 24-bit Timer/Counters and one 16-bit Timer/Counter for Control (TCC), with extended functions: <ul style="list-style-type: none"> Up to four compare channels with optional complementary output Generation of synchronized pulse width modulation (PWM) pattern across port pins Deterministic fault protection, fast decay and configurable dead-time between complementary outputs Dithering that increase resolution with up to 5 bit and reduce quantization error
RTC	<ul style="list-style-type: none"> 32-bit Real Time Counter (RTC) with clock/calendar function 	<ul style="list-style-type: none"> 32-bit Real Time Counter (RTC) with clock/calendar function 	<ul style="list-style-type: none"> 32-bit Real Time Counter (RTC) with clock/calendar function
WDT	<ul style="list-style-type: none"> Watchdog Timer (WDT) 	<ul style="list-style-type: none"> Watchdog Timer (WDT) 	<ul style="list-style-type: none"> Watchdog Timer (WDT)
USB / CAN	<ul style="list-style-type: none"> One full-speed (12Mbps) Universal Serial Bus (USB) 2.0 interface Embedded host and device function Eight endpoints 	<ul style="list-style-type: none"> One full-speed (12Mbps) Universal Serial Bus (USB) 2.0 interface Embedded host and device function Eight endpoints 	<ul style="list-style-type: none"> Up to two Controller Area Network (CAN) interfaces: <ul style="list-style-type: none"> CAN 2.0A/B CAN-FD 1.0 Each CAN interface have two selectable pin locations to switch between two external CAN transceivers (without the need for an external switch)
SERCOM	<ul style="list-style-type: none"> Up to six Serial Communication Interfaces (SERCOM), each configurable to operate as either: <ul style="list-style-type: none"> USART with full-duplex and single-wire half-duplex configuration I2C up to 3.4MHz SPI LIN slave 	<ul style="list-style-type: none"> Up to six Serial Communication Interfaces (SERCOM) including one low-power SERCOM, each configurable to operate as either: <ul style="list-style-type: none"> USART with full-duplex and single-wire half-duplex configuration I2C up to 3.4MHz SPI LIN slave 	<ul style="list-style-type: none"> Up to six Serial Communication Interfaces (SERCOM), each configurable to operate as either: <ul style="list-style-type: none"> USART with full-duplex and single-wire half-duplex configuration I2C up to 3.4MHz SPI LIN master/slave RS-485
I2S	<ul style="list-style-type: none"> One two-channel Inter-IC Sound (I2S) interface 		
ADC	<ul style="list-style-type: none"> One 12-bit, 350ksps Analog-to-Digital Converter (ADC) with up to 20 channels 	<ul style="list-style-type: none"> One 12-bit, 1MSPS Analog-to-Digital Converter (ADC) with up to 20 channels 	<ul style="list-style-type: none"> Two 12-bit, 1Msps Analog-to-Digital Converter (ADC) with up to 12 channels each

	<ul style="list-style-type: none"> Differential and single-ended input 1/2x to 16x programmable gain stage Automatic offset and gain error compensation Oversampling and decimation in hardware to support 13-, 14-, 15- or 16-bit resolution 	<ul style="list-style-type: none"> Differential and single-ended input Automatic offset and gain error compensation Oversampling and decimation in hardware to support 13-, 14-, 15-, or 16-bit resolution Can use with 3 OPAMPs 	(20 unique channels in total) <ul style="list-style-type: none"> Differential and single-ended input Automatic offset and gain error compensation Oversampling and decimation in hardware to support 13-, 14-, 15- or 16-bit resolution One 16-bit Sigma-Delta Analog-to-Digital Converter (SDADC) with up to 3 differential channels
DAC	<ul style="list-style-type: none"> 10-bit, 350ksps Digital-to-Analog Converter (DAC) 	<ul style="list-style-type: none"> Two 12-bit, 1MSPS Dual Output Digital-to-Analog Converter (DAC) 	<ul style="list-style-type: none"> 10-bit, 350ksps Digital-to-Analog Converter (DAC)
Comparators	<ul style="list-style-type: none"> Two Analog Comparators (AC) with window compare function 	<ul style="list-style-type: none"> Two Analog Comparators (AC) with window compare function 	<ul style="list-style-type: none"> Four Analog Comparators (AC) with window compare function <50ns
Custom Logic		<ul style="list-style-type: none"> One Configurable Custom Logic (CCL) 	<ul style="list-style-type: none"> One Configurable Custom Logic (CCL)
PTC	<ul style="list-style-type: none"> Peripheral Touch Controller (PTC) 256-Channel capacitive touch and proximity sensing 	<ul style="list-style-type: none"> Peripheral Touch Controller (PTC) 169-Channel capacitive touch and proximity sensing Wake-up on touch in standby mode 	<ul style="list-style-type: none"> Peripheral Touch Controller (PTC) 256-Channel capacitive touch and proximity sensing DMA
I/O	<ul style="list-style-type: none"> Up to 52 programmable I/O pins 	<ul style="list-style-type: none"> Up to 51 programmable I/O pins 	<ul style="list-style-type: none"> Up to 52 programmable I/O pins
Other	<ul style="list-style-type: none"> CRC-32 generator 	<ul style="list-style-type: none"> CRC-32 generator Three Operational Amplifiers (OPAMP) One AES encryption engine One True Random Generator (TRNG) 	<ul style="list-style-type: none"> CRC-32 generator Frequency Meter Integrated Temperature Sensor Hardware Divide and Square Root Accelerator (DIVAS)

Development Software Matrix

Environment	Windows	Linux	OS X
Standalone (Test program coming soon)	SAM-BA bootloader , SAM-BA bootloader utility , Test program	SAM-BA bootloader , SAM-BA bootloader utility , Test program	SAM-BA bootloader , SAM-BA bootloader utility
Arduino	Arduino core files includes SAM-BA bootloader and bootloader utility, Test sketch	Arduino core files includes SAM-BA bootloader and bootloader utility, Test sketch	Arduino core files includes SAM-BA bootloader and bootloader utility, Test sketch
Atmel Studio (coming soon)	Test program project files	NA	NA
PlatformIO	Summer 2017?	Summer 2017?	Summer 2017?

Pinout

Arduino Pinout

===== MattairTech MT-D21E rev B (ATSAMx21Exxx) =====												
Other	COM	PWM	Analog	INT	Arduino*	Arduino*	INT	PWM	COM	Other		
=====												
XI32(+)					A0	RST				BOOT(+)		
X032(+)					A1	Gnd						
DAC			*		2	A2	Vbat					
REFA			*		3	A3	A31	31	*	IO/B(+)		
REFB			*	*	4	A4	A30	30	*	CLK(+)		
DAC1(L)			*	*	5	A5	NC					
LED(+)		TCC10	*		6	A6	A28 (D/C)	28	*			
VM		TCC11	*		7	A7	A27	27	*	A/CS(+M)		
		TCC00	*	NMI	8	A8	A23	23	*	TC41/TC01~ SS		
		TCC01	*	*	9	A9	A22	22	*	TC40/TC00~ MISO(+M)		
	TX1	TCC02	*		10	A10	A19	19	*	SCK(+M)		
	RX1	TCC03	*		11	A11	A18	18	*	MOSI(+M)		
	TX2	TC30/TC40~		*	14	A14	A17	17	*	TCC21 SCL(+)		
	RX2	TC31/TC41~			15	A15	A16	16	*	TCC20 SDA(+)		
M=Memory device installed					NC	NC						
					NC	NC						
					Vbus	3.3V						
USB D- (D/L)(+), CAN TX (C)	TC50	24			A24	Vcc						
USB D+ (D/L)(+), CAN RX (C)	TC51	25			A25	Vin						
Chip Variant:					Gnd	USB	Gnd					
D=D21, L=L21, C=C21												

! Vcc is 3.3V by default.
DO NOT exceed 3.6V on Vcc or
any IO pin with the D21 or
L21 installed. 5V is allowed
ONLY with the C21 installed.

* Most pins can be used for more than one function. The same port pin number printed on the board is also used in Arduino (without the 'A') for all of the supported functions (ie: digitalWrite(), analogRead(), analogWrite(), attachInterrupt(), etc.).

+ This alternate function is enabled by default (+M functions enabled only when a memory device is installed). Thus, the associated header pin cannot be used. Solder jumpers can be used to enable or disable the alternate onboard function.

~ When two timers are shown, the second is for L21/C21. TC5 is TC1 on the L21/C21.

Silkscreen Legend:

Top: A circled pin means analog function and '*' means alternate function (see + above)
Bottom: A circled pin means analog function

Main Header Pins (Power)

Pin	Description
Gnd (3)	There are three ground pins.
Vbus	<p>Vbus is connected directly to the Vbus pin (5V) of the USB connector. It is routed through a schottky diode and through J6 to the regulator input circuitry, which includes a 250mA (hold) PTC fuse and a 4.7uF ceramic capacitor.</p> <p>Vbus voltage can be measured on pin A7 by connecting J12 and setting J3 (3-pad) away from the circle mark. This will form a resistor divider with 200Kohm connected to Vbus, and 20Kohm connected to ground. Pin A7 will be pulled to near ground level when Vbus is disconnected (there will be a small voltage because of diode leakage).</p> <p>If the C21 is installed, then Vbus can be connected directly to Vcc by setting jumper X1 and changing J5 (3-pad) to the position away from the asterisk. Vcc will then operate at ~5V from USB Vbus. Do not set X1 if the D21 or L21 is installed.</p>
Vin	<p>Up to 16V can be connected to the Vin pin. It is routed through a schottky diode to the regulator input circuitry, which includes a 250mA (hold) PTC fuse and a 4.7uF ceramic capacitor. The schottky diode can be shorted with J1, eliminating the voltage drop across the diode (useful for battery applications). Note that when the diode is shorted, reverse-polarity protection is disabled, and J6 should be disconnected to prevent Vbus current from flowing into Vin. The PTC fuse provides a small resistance, which helps control inductive voltage spiking that can result from the combination of low input capacitor ESR and inductive input wiring. Avoid excessive inductance on Vin.</p> <p>Vin voltage can be measured on pin A7 by connecting J12 and setting J3 (3-pad) toward the circle mark. This will form a resistor divider with 200Kohm connected to Vin, and 20Kohm connected to ground. Pin A7 will be pulled to near ground level when Vin is disconnected (there will be a small voltage because of diode leakage).</p>
Vcc	<p>This pin is connected to the Vcc and VddAna (through a ferrite bead) pins on the microcontroller, the Cortex debug header Vcc pin, the reset pullup, and the TWI pullup resistors, as well as the memory device Vcc pin and CS pullup (if installed). By default, Vcc is connected to 3.3V through J5 (3-pad), which in turn is connected to the output of the onboard regulator. The Vcc pin can also be used as an input. Disconnect J5 to supply power from an external source to the Vcc pin.</p> <p>If the C21 is installed, then Vbus can be connected directly to Vcc by setting jumper X1 and changing J5 (3-pad) to the position away from the asterisk. Vcc will then operate at ~5V from USB Vbus. Do not set X1 if the D21 or L21 is installed.</p>
3.3V	3.3V is connected to the output of the onboard 3.3V regulator, which has a 10uF ceramic capacitor installed. By default, Vcc is connected to 3.3V through J5 (3-pad).
Vbat	The Vbat pin can connect to the SRAM Vbat pin (see J22) and/or Vcc (through a schottky diode, see J23). A coin cell or other battery (up to 3.5V) can be connected. When using with Vcc, care must be taken to limit current.

CAUTION

Higher regulator input voltages mean larger voltage drops and thus higher thermal dissipation for a given amount of current. Be sure to limit current consumption to prevent excessive heat when using higher voltages and/or currents. The regulator will enter thermal shutdown if it gets too hot. Note that the PTC fuse is located near the regulator, so high temperatures will lower the PTC trip and hold currents.

Main Header Pins (Signal)

Pin		Description
A0 (XI32), A1 (XO32)	+	By default, these pins are disconnected, as jumpers J16 and J17 are set to route A0 and A1 to the onboard 32.768KHz crystal. Alternatively, these can be used for digital functions by switching J16 and J17.
A2 (DAC)	O	This can be used for analog or digital functions. Pin A2 can be used as a DAC output.
A3 (REFA), A4 (REFB)	O	These can be used for analog or digital functions. Alternatively, jumpers J19 and/or J18 can be set to enable both a 100nF capacitor and a 1uF capacitor for use with an external voltage reference.
A5	O	This can be used for analog or digital functions.
A6 (LED)	+ O	By default, this pin is connected to a blue LED through jumper J14 and a 1Kohm resistor. The LED circuit should consume around 0.5mA. Drive the pin high to turn on the LED. Alternatively, this pin can be used for digital functions by clearing J14.
A7 (VM) Voltage Measure	O	This can be used for analog or digital functions. Alternatively, this pin can be connected to the voltage divider for measurement of Vin or Vbus by setting J3 and J12 appropriately (see jumper settings).
A8 - A11	O	These can be used for analog or digital functions.
A14 (XIN), A15 (XOUT)		These can be used for digital functions. Pin A14 can be used with an external clock. Alternatively, jumpers J10 and J11 can be switched to route A15 and A14 to the onboard 16MHz crystal.
A24 (USB D-, CAN TX with C21), A25 (USB D+, CAN RX with C21)	+	These can be used for digital functions. The MCU pins A24 and A25 are always connected to the main header. By default, with the D21 and L21, these pins are also connected to pins D- and D+ of the USB connector (and to the ESD device) through jumpers J7 and J4. These header pins, along with the adjacent Vbus and Ground pins can be used for a panel-mount USB connector.
A16 (SDA), A17 (SCL)	+	By default, jumpers J8 and J9 are enabled, which connects two 4.7Kohm pullup resistors for use with I2C. Alternatively, these pins can

		be used for digital functions by clearing J8 and J9.
A18 (MOSI), A19 (SCK), A22 (MISO)	+M	When the optional memory device is installed, these pins connect to it using a SPI bus. Otherwise, they can be used for digital functions.
A23		This can be used for digital functions.
A27 (A/CS) Jumper A / MEM CS	+ +M	By default, this pin is connected to Jumper A (and to the optional memory device CS pin) through jumper J13. The pin is grounded when the jumper is installed. If a memory device is present, then there will also be a 100K pullup resistor. Do not use the header pin when either Jumper A or a memory device is installed. Alternatively, this pin can be used for digital functions by clearing J14.
A28 (NC with L21)	+	This can be used for digital functions.
A30 (CLK) SWD CLK	+	This can be used for digital functions. Additionally, this pin is connected to the Cortex debug header where it is used as SWD CLK. It is important to leave this pin floating or pulled up to Vcc after reset, as it is used to detect debugger probes. If operating in noisy environments, it is recommended to connect an external 1Kohm resistor to Vcc.
A31 (IO/B) SWD IO / Button B	+	This can be used for digital functions. Additionally, the pin is routed to the Cortex debug header where it is used as SWD IO. Alternatively, this pin can be connected to Button B through jumper J15 (note that this button can also be used for RST). This button is debounced using a 1.8Kohm resistor and a 100nF capacitor. The pin is brought to ground when the button is pressed.
RST (BOOT) Bootloader entry	+	RST connects to the reset pin of the microcontroller, to Button B through jumper J15 (note that this button can also be for general purpose use; see A31 above), to the Cortex debug header, and to a 10K pullup resistor to Vcc. When the button is connected to RST, it can be pressed twice in quick succession to enter the bootloader.
NC		These pins are not connected. There are 5 pins marked NC.
Cortex Debug Header		This 10-pin, 50-mil header can be connected to an external programmer/debugger. Be sure that nothing is connected to main header pins A30 and A31 (other than a pullup on A30).

+ = By default, header pin in use by alternate onboard function (use solder jumpers to change)

+M = pin in use only when optional memory device is installed

O = analog functions available

Solder Jumpers



*Silkscreen Legend: * = Set by default, o = Unset by default, ? = Depends on hardware options. The symbol also serves as a polarity indicator for 3-pad jumpers, referenced in the table below.*

Jumper	Description
J1: Vin diode disable O Default: NC	Up to 16V can be connected to the Vin pin. It is routed through a schottky diode to the regulator input circuitry. The schottky diode can be shorted with J1, eliminating the voltage drop across the diode (useful for battery applications). Note that when the diode is shorted, reverse-polarity protection is disabled, and J6 should be disconnected to prevent Vbus current from flowing into Vin.
J2: USB Shield Ground O Default: NC	Jumper J2 can be closed to connect the USB shield to ground. The USB specification calls for the USB shield to be connected to ground on the host side only. However, some prefer to have it grounded. Bear in mind that the USB shield will then act as an antenna. To avoid this, an 0603 component and an 0402 (ie: 1Mohm resistor and 4.5nF capacitor) may be soldered on the pads.
J3: Divider to Vin / Vbus 3-pad: O = Vin O Default: NC	Vin or Vbus voltage can be measured on pin A7 by connecting J12 and setting J3 toward the circle mark (for Vin) or away from the circle mark (for Vbus). This will form a resistor divider with 200Kohm connected to Vin, and 20Kohm connected to ground. Pin A7 will be pulled to near ground level when Vin/Vbus is disconnected (there will be a small voltage because of diode leakage).
J4: A25 to USB D+ ? Default depends: D21 or L21: Connected C21: NC	Microcontroller pins A24 and A25 are always connected to header pins A24 and A25. By default, with the D21 and L21, these pins are also connected to the D- and D+ pins of the Mini USB connector (and to the ESD device) through jumpers J7 and J4. The header pins, along with the adjacent Vbus and Ground pins can be used for a panel-mount USB connector. Leave unconnected with the C21.

J5: Vcc to 3.3V / Vbus !! Use Extra Caution 3-pad: * = 3.3V * Default: 3.3V	By default, this connects Vcc to the 3.3V regulator output. Alternatively, if the C21 is installed, switch to the position away from the asterisk to connect Vcc to USB Vbus and operate at ~5V. Additionally, X1 must be connected as well. DO NOT CONNECT Vcc to Vbus with the D21 or L21 installed, or with the FLASH installed. Leave J5 completely unconnected (all 3 pads) if supplying a regulated voltage (3.6V or less for the D21 and L21) externally on the Vcc pin.
J6: Vbus Power * Default: Connected	This connects Vbus to the regulator input circuitry. Two schottky diodes, one for Vin and one for Vbus, facilitate automatic power switching. If only Vin is used, or if J1 is connected, disconnect J6 to prevent Vbus from being used.
J7: A24 to USB D- ? Default depends: D21 or L21: Connected C21: NC	See J4.
J8: SDA Pullup Resistor * Default: Connected	This connects pin A16 through a 4.7Kohm resistor to Vcc for use with I2C SDA.
J9: SCL Pullup Resistor * Default: Connected	This connects pin A17 through a 4.7Kohm resistor to Vcc for use with I2C SCL.
J10: A15 to 16MHz Crystal / Header Pin J11: A14 to 16MHz Crystal / Header Pin 3-pad: * = Header Pin * Default: Header Pin	J10 and J11 determine whether microcontroller pins A15 and A14 connect to header pins A15 and A14 or to the 16MHz crystal. By default, the header pins are connected.
J12: VM Enable O Default: NC	See J3.
J13: A/CS Enable Jumper A / MEM CS * Default: Connected	This connects pin A27 through a 1.8Kohm resistor to Jumper A. The pin is grounded when the jumper is installed. This also connects pin A27 to the optional memory device CS pin. If a memory device is installed, there will be a 100K pullup as well. Be sure to remove Jumper A if a memory device is installed.
J14: LED Enable * Default: Connected	This jumper connects pin A6 to a blue LED through a 1Kohm resistor. Drive the pin high to turn on the LED. The LED circuit consumes ~0.5mA@3.3V.
J15: Button B Selection 3-pad: * = RST * Default: RST	This connects Button B to either the RST pin, which is the default, or to pin A31 for general purpose use. This button is debounced using a 1.8Kohm resistor and a 100nF capacitor. The pin is grounded when the button is pressed. Note that pin A31 is also used by the Cortex debug header (SWD IO). The button can be completely disconnected by removing solder from all three pads.
J16: A0 to 32.768KHz Crystal / Header Pin J17: A0 to 32.768KHz Crystal / Header Pin	J16 and J17 determine whether microcontroller pins A0 and A1 connect to header pins A0 and A1 or to the 32.768KHz crystal. By default, they connect to the crystal.

3-pad: * = Crystal * Default: Crystal	
J18: REFB capacitors O Default: NC	When using pin A4 as REFB, close J18 to enable both a 100nF capacitor and a 1uF capacitor from A4 to ground.
J19: REFA capacitors O Default: NC	When using pin A3 as REFA, close J19 to enable both a 100nF capacitor and a 1uF capacitor from A3 to ground.
J20: A28/VDDCORE(L) to A28 / VSW INDUCTOR ! Use Caution 3-pad: ? = A28 ? Default depends: D21 or C21: A28 L21: VSW INDUCTOR	D21 or C21 installed: This jumper connects the A28 MCU pin to header pin A28. Since there is no internal buck converter with the D21 or C21, there is a 0 ohm resistor installed instead of the inductor. Do not change this jumper. L21 installed: This jumper connects the VDDCORE pin (pin A28 does not exist with the L21) to a 10uH inductor to VSW. It will also connect to both a 1uF capacitor and a 100nF capacitor to ground. This supports the internal buck converter of the L21. Do not change this jumper.
J21: MEM _WP_ to Gnd ? Default depends: SRAM: NC FLASH: NC EEPROM: Vcc NO MEMORY: NC	SRAM installed: The SRAM pin is NC, so leave unconnected. FLASH installed: This jumper connects the optional FLASH _WP_ pin to Gnd. Firmware must also enable write protection, thus writes can still be made with _WP_ low. If J21 is not soldered, _WP_ will be pulled high by an internal resistor. EEPROM installed: This jumper connects the optional EEPROM _WP_ pin to Gnd. Firmware must also enable write protection, thus writes can still be made with _WP_ low. If J21 is not soldered, _WP_ will float, so leave it connected. No memory device installed: Leave unconnected
J22: MEM to Vbat / Vcc 3-pad: ? = Vbat ? Default depends: SRAM: Vbat FLASH: Vcc EEPROM: Vcc NO MEMORY: NC	SRAM installed: This jumper connects the optional SRAM Vbat pin to the header Vbat pin (solder the ? Side). FLASH or EEPROM installed: This jumper connects the optional FLASH or EEPROM _HOLD_ pins to Vcc (solder opposite the ?). If the Vbat pin is not used for other purpose (not connected to Vcc), then the hold feature can be used by connecting _HOLD_ to Vbat, then connecting the Vbat pin to any free GPIO pin. No memory device installed: Leave unconnected
J23: Vbat to Vcc ! Use Caution O Default: NC	This jumper connects the Vbat pin to Vcc through a schottky diode. The maximum voltage allowed on Vbat is 3.5V. The diode has a voltage drop of ~0.25V typ. (~0.4V max.) and a reverse leakage of ~0.1uA (~10uA @ 85C). As Vbat is intended for a coin cell, great care must be taken to limit current consumption both to prevent excessive voltage drop (lithium coin cells have a relatively high internal resistance) and premature battery depletion.
X1: Vbus to Vcc !! Use Extra Caution O Default: NC	This jumper connects USB Vbus to Vcc, so that Vcc will operate at ~5V. Do this only if the C21 is installed. DO NOT CONNECT with the D21 or L21 installed, or with the FLASH installed. In addition, jumper J5 will have to be switched to the side near the asterisk mark to complete the connection from Vbus to Vcc.

Memory Device

A serial memory device can optionally be installed, with the choice of 128KB SRAM, 512KB FLASH, or 64KB EEPROM. The device is connected to the microcontroller via SPI pins 18, 19, 22, and 27 (CS). All lines are also routed to the main header pins, though CS can be disconnected from the main header by clearing solder jumper J13 (by default, this jumper is connected). Note that due to limitations of the SPI peripheral, 12MHz is the maximum clock speed. All devices are compatible with Mode 0. Be sure to remove Jumper A if a memory device is installed, as this line is shared with CS (the memory device will still work though).

Arduino libraries coming soon.

128KB SRAM with Battery Backup

Features (from Microchip [datasheet](#))

- Microchip Technology Inc. 23LCV1024
- 2.5V – 5.5V Supply Voltage Range
- External Battery Backup Support from Vin Pin
- Read Current: 3 mA at 5.5V, 20 MHz
- Standby Current: 4 uA at +85°C
- Unlimited Read and Write Cycles, Zero Write Time
- 128K x 8-bit Organization: 32-byte page
- Byte, Page and Sequential mode for Reads and Writes

Hardware Settings

- Solder Jumper J22 (3-pad) must be set to Vbat (pad near ?)
 - A coin cell up to 3.5V may be connected to the Vbat main header pin
 - When Vcc is less than about 1.8V, switchover to the battery will occur (drawing ~1uA)
 - SRAM contents will be kept with Vbat as low as about 1V
- Solder Jumper J21 must be cleared (the associated 23LCV1024 pin is NC)

512KB FLASH

Features (from Adesto [datasheet](#))

- Adesto® AT25XV041B
- 1.65V – 4.4V, **Do not operate at 5V** (the C21 supports 5V)
- Uniform 4-Kbyte, 32-Kbyte, or 64-Kbyte Block Erase; Full Chip Erase
- Hardware Controlled Locking of Protected Sectors via WP Pin
- 128-byte, One-Time Programmable (OTP) Security Register
 - 64 bytes factory programmed with a unique identifier, 64 bytes user programmable

- Byte/Page Program (1 to 256 Bytes)
- Dual-Input Byte/Page Program (1 to 256 Bytes)
- Sequential Program Mode Capability
- 1.85ms Typical Page Program (256 Bytes) Time
- 45ms Typical 4-Kbyte Block Erase Time (360ms for 32KB and 720ms for 64KB)
- Automatic Checking and Reporting of Erase/Program Failures
- 200nA Ultra Deep Power Down current (Typical), 5µA Deep Power-Down, 25uA Standby
- 3.5mA Active Read Current (Typical)
- Endurance: 100,000 Program/Erase Cycles
- Data Retention: 20 Years

Hardware Settings

- Solder Jumper J22 (3-pad) must be set to Vcc (pad away from ?) which will disable `_HOLD_`
 - If Vbat is unused for other purposes, J22 can be connected to Vbat and the Vbat header pin can then be connected to any free GPIO, allowing control of `_HOLD_`
- Solder Jumper J21 should be set, which will enable `_WP_`
 - Firmware must also enable write protection, thus writes can still be made with `_WP_` low
 - If J21 is not soldered, `_WP_` will be pulled high by an internal resistor

64KB EEPROM

Features (from ON Semiconductor [datasheet](#))

- ON Semiconductor CAT25512
- 1.8V to 5.5V Supply Voltage Range
- SPI Modes (0,0) & (1,1)
- 128-byte Page Write Buffer
- Additional Identification Page with Permanent Write Protection
- Self-timed Write Cycle
- Hardware and Software Protection
- Block Write Protection, Protect 1/4, 1/2 or Entire EEPROM Array
- Low Power CMOS Technology
- 4,000,000 Program/Erase Cycles
- 200 Year Data Retention

Hardware Settings

- Solder Jumper J22 (3-pad) must be set to Vcc (pad away from ?) which will disable `_HOLD_`
 - If Vbat is unused for other purposes, J22 can be connected to Vbat and the Vbat header pin can then be connected to any free GPIO, allowing control of `_HOLD_`
- Solder Jumper J21 must be set, which will enable `_WP_`
 - Firmware must also enable write protection, thus writes can still be made with `_WP_` low
 - J21 must not be cleared. There is no internal pullup resistor, so the `_WP_` pin will float.

Power Supply

Features

- *Microchip MCP1703-3302E/DB (from Microchip [datasheet](#))*
 - Onboard 3.3V, 250mA LDO regulator
 - Supports up to 16V DC input on Vin pin
 - Extremely low quiescent current (2.0uA typical)
 - Over-current and over-temperature protection
 - low dropout (525mV typical @ 250mA, 725mV max. @ 250mA)
 - 0.4% output tolerance typical
 - 4.7uF, 25V X7R ceramic input capacitor, 10uF X7R ceramic output capacitor
- *Powered by USB Vbus or external power source on Vin (up to 16V)*
 - Simple diode power source switching, reverse-polarity protection
 - Vbus can be disconnected. Vin diode can be bypassed.
- *PTC resettable fuse (250mA hold / 500mA trip)*
 - If tripped, the PTC fuse will auto-reset when the over-current condition is removed.
 - PTC fuse is near the regulator, so high temperatures will lower the trip and hold currents.
 - The PTC fuse provides a small resistance, which helps control inductive voltage spiking that can result from the combination of low input capacitor ESR and inductive input wiring. Avoid excessive inductance on Vin.
- *Vbat to Vcc (advanced use)*
 - The Vbat pin, which supports $\leq 3.5V$, can be connected to Vcc through a schottky diode.
 - As Vbat is intended for a coin cell, great care must be taken to limit current consumption both to prevent excessive voltage drop (lithium coin cells have a relatively high internal resistance) and premature battery depletion. Vbat can also be connected to the SRAM.
- *5V operation (C21 only)*
 - With the C21 installed, USB Vbus can be connected to Vcc for 5V operation
 - With 5V operation, the regulator is not used (can still use externally on 3.3V pin)
- *On-chip buck converter (L21 only)*
 - If the L21 is installed, an inductor will replace the 0-ohm resistor, and along with 1uF and 100nF capacitors, it supports the on-chip buck converter for increased power efficiency.

See Solder Jumper section and Header Pins sections for configuration details.

CAUTION

Do not operate the D21, L21, or FLASH memory at 5V

Other Hardware

- **Blue LED**
 - PORT pin A6, connected through jumper J14. Drive high to turn on.
 - ~0.5mA @ 3.3V current consumption
- **Jumper A**
 - PORT pin A27, connected through jumper J13. Install jumper to ground A27.
 - 1.8Kohm series resistor installed. Leave jumper off when memory device is installed.
- **Reset Button / Button B**
 - Connected to reset by default using jumper J15. The reset button can be pressed twice in quick succession to enter the SAM-BA bootlader. Button can connect to PORT pin A31 instead by switching J15.
 - 1.8Kohm series resistor and 100nF capacitor installed for debouncing.
- **Coin Cell Input**
 - The Vbat pin (<3.5V) can connect to SRAM Vbat and/or Vcc (through schottky diode).
 - When using with Vcc, care must be taken to limit current (advanced users).
- **Cortex Header**
 - 10-pin, 50-mil header pre-installed, for programming and debugging.
- **Crystals**
 - 32.768KHz with two 18pF capacitors connected through jumpers J16 and J17.
 - 16MHz with two 13pF capacitors connected through jumpers J10 and J11.
- **I2C Pullups**
 - 4.7Kohm each (SDA, SCL) to Vcc through jumpers J8 and J9.
- **Mini USB Connector**
 - ESD protection on D+, D-, and Vbus. D+/D- pullup will not trigger false Vbus detection.
 - Vbus, D-, D+, and Gnd available on main header for panel mount connector use.
- **Voltage Measurement**
 - Simple voltage divider with 200Kohm connected to either Vin or USB Vbus using jumper J3, and 20Kohm connected to ground through J12, which MUST be installed.
 - Connected to PORT pin A7. Be sure to use a sufficiently long sampling time.
- **Analog Reference Filters**
 - J19 (A3) and/or J18 (A4) can be set to enable both a 100nF capacitor and a 1uF capacitor so that the pin can be used with an external voltage reference.

The schematic diagram illustrates the internal components and connections of the MT-D21E development board. Key components include the ATSAM21E microcontroller, 128KB Battery Backed SRAM, 512KB FLASH, and 64KB EEPROM. The board features a 3.3V regulator, a 3.3V USB-to-UART bridge, and a 3.3V USB-to-UART bridge. The schematic shows the connection of various components like resistors, capacitors, and connectors. It also includes a header legend and a jumpers defaults section.

Header Legend:

- Alternate onboard function may, depending on options, be in use by default, so the header pin is unavailable.
- Analog peripheral available

Jumpers Defaults:

- Set
- Unset
- Depends

The symbol indicates polarity with 3-pad jumpers. The location of the symbol on the schematic corresponds to the PCB location.

Parts List

Note that part designators are not yet used, so use this table along with the schematic.

<i>Part #</i>	<i>Description</i>	<i>Temp.</i>	<i>Qty</i>	<i>Notes</i>
ATSAMD21E17A	IC MCU 32BIT 128KB FLASH 32TQFP	-40 °C ~ 85 °C	1	
ATSAMD21E18A	IC MCU 32BIT 256KB FLASH 32TQFP	-40 °C ~ 85 °C	1	
ATSAMC21E18A	IC MCU 32BIT 256KB FLASH 32TQFP	-40 °C ~ 85 °C	1	
ATSAML21E18B	IC MCU 32BIT 256KB FLASH 32TQFP	-40 °C ~ 85 °C	1	
7A-16.000MAHE-T	CRYSTAL 16.000 MHZ 12PF SMD	-40 °C ~ 85 °C	1	
GRM1555C1H130GA01D	13pF $\pm 2\%$ 50V Ceramic Capacitor C0G, NP0 0402	C0G, -55 °C ~ 125 °C	2	
9HT7-32.768KDZF-T	CRYSTAL 32.768KHZ 12.5PF SMD	-40 °C ~ 85 °C	1	
GRM1555C1H180FA01D	18pF $\pm 1\%$ 50V Ceramic Capacitor C0G, NP0 0402	C0G, -55 °C ~ 125 °C	2	
LI0603G221R-10	FERRITE 700MA 220 OHM 0603 SMD	-40 °C ~ 85 °C	1	
C1608X7S1A475K080AC	CAP CER 4.7UF 10V 10% X7S 0603	X7S, -55 °C ~ 125 °C	1	
GRM155R71C104KA88D	CAP CER .1UF 16V X7R 0402	X7R, -55 °C ~ 125 °C	5	
LMK107B7105KA-T	CAP CER 1.0UF 10V X7R 0603	Industrial	4	
VLS201612CX-100M	FIXED IND 10UH 770MA 540 MOHM	-40 °C ~ 105 °C	1	L21 only
ERJ-6GEY0R00V	RES SMD 0.0 OHM JUMPER 1/8W 0805	-55 °C ~ 155 °C	1	D21 or C21
ERJ-3EKF1002V	10K, thick film, 1/10W, 1%	-55 °C ~ 125 °C	1	
USB Connector	Type Mini AB - 5 pin		1	
TPD4E1U06DCKR	TVS DIODE 5.5VWM 15VC SC70-6	-40 °C ~ 85 °C	1	
Tactile Switch	SWITCH TACTILE SPST-NO		1	
ERJ-3EKF1801V	RES SMD 1.8K OHM 1% 1/10W 0603	-55 °C ~ 125 °C	1	
GRM155R71C104KA88D	CAP CER .1UF 16V X7R 0402	X7R, -55 °C ~ 125 °C	1	
Jumper pins	CONN HEADER 36-40PS .100 STR GOLD		2	
SPC02SXCN-RC	2 (1 x 2) Position Shunt Connector Open Top 0.100" (2.54mm) Gold		1	
ERJ-3EKF1801V	RES SMD 1.8K OHM 1% 1/10W 0603	-55 °C ~ 125 °C	1	

<i>Part #</i>	<i>Description</i>	<i>Temp.</i>	<i>Qty</i>	<i>Notes</i>
MCP1703-3302E/DB	IC REG LDO 3.3V 0.25A SOT223-3	-40°C ~ 125°C	1	
TMK212AB7475KG-T	CAP CER 4.7UF 25V 10% X7R 0805	X7R, -55°C ~ 125°C	1	
CL21B106KOQNNNE	CAP CER 10UF 16V X7R 0805	X7R, -55°C ~ 125°C	1	
0ZCJ0025FF2E	PTC Resettable Fuse 16V 250mA lh Surface Mount 1206 (3216 Metric), Concave	Industrial	1	
RB160MM-40TR	DIODE SCHOTTKY 40V 1A PMDU	-50°C ~ 125°C	2	
BAT30KFILM	DIODE SCHOTTKY 30V 300MA SOD523	-55°C ~ 150°C	1	
ERJ-3EKF4701V	RES 4.7K OHM 1/10W 1% 0603 SMD	-55°C ~ 125°C	2	
ERJ-3EKF2003V	RES 200K OHM 1/10W 1% 0603 SMD	-55°C ~ 125°C	1	
ERJ-3EKF2002V	20K, thick film, 1/10W, 1%	-55°C ~ 125°C	1	
LB Q39G-L2OO-35-1	LED CHIPLED BLUE 470NM 0603 SMD	-40°C ~ 85°C	1	
ERJ-3EKF1001V	RES SMD 1K OHM 1% 1/10W 0603	-55°C ~ 125°C	1	
23LCV1024-I/ST	IC NVSRAM 1MBIT 20MHZ 8TSSOP	-40°C ~ 85°C (TA)	1	optional
CAT25512YI-GT3	IC EEPROM 512KBIT 20MHZ 8TSSOP	-40°C ~ 85°C (TA)	1	optional
AT25XV041B-XMHV-T	IC FLASH MEM SPI 4MBIT 8TSSOP	-40°C ~ 85°C (TC)	1	optional
ERJ-3EKF1003V	RES 100K OHM 1/10W 1% 0603 SMD	-55°C ~ 125°C	1	optional
C1005X7R1C104K050BC	CAP CER .1UF 16V X7R 0402	X7R, -55°C ~ 125°C	1	optional
Main headers	CONN HEADER 36-40PS .100 STR GOLD		40	
3220-10-0100-00	BOX HEADER, 0.050 10 POS		1	

Dimension Drawing

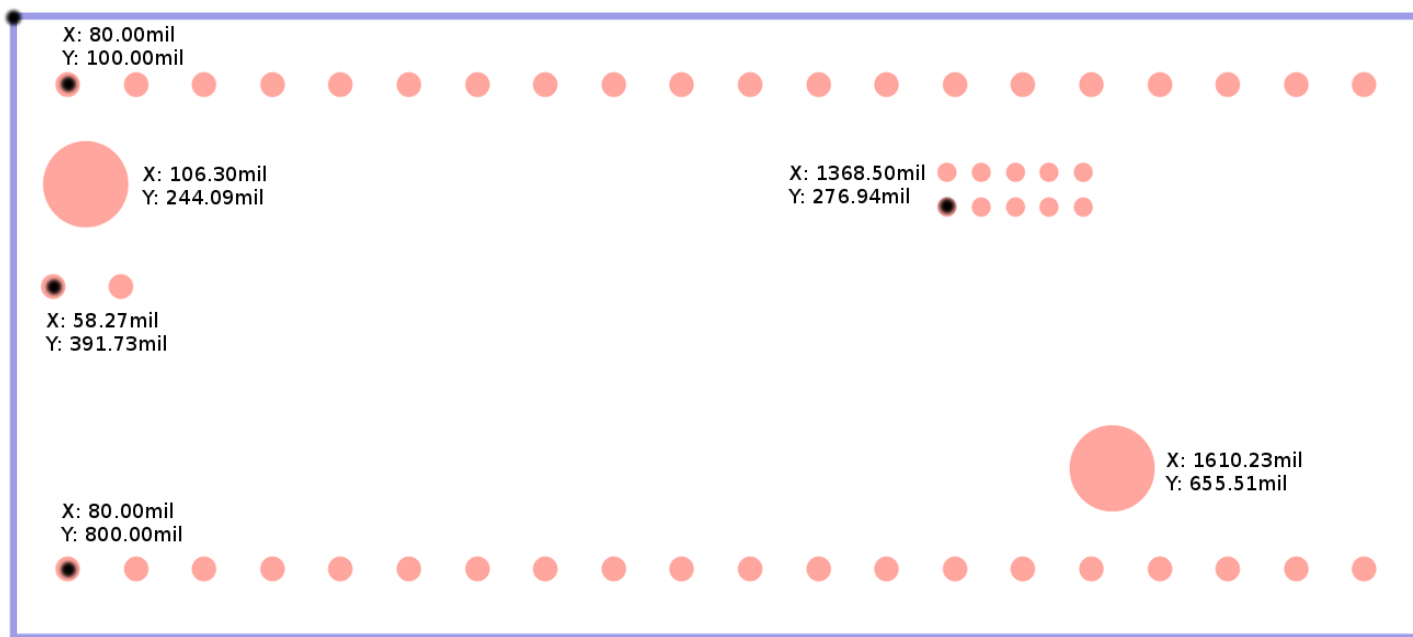
MT-D21E rev B PCB Dimensions

Download Excellon drill file from http://www.mattairtech.com/docs/MT-D21E/MT-D21E_header_and_mounting_holes.cnc

X: 0.00mil
Y: 0.00mil

Board Dimensions: 2060.0mil wide, 900.0mil high

Mounting holes: 125mil diameter hole, 32.5mil annulus



All pins within each header are on 100mil spacing. The two main headers are on a common 100mil grid.
Note that both the 2 jumper pins and the Cortex header are NOT on this same grid.

MattairTech Arduino SAM M0+ Core

Please visit <https://github.com/mattairtech/ArduinoCore-samd> for updated documentation and information on the new 1.6.8-beta release with support for OS X and many updates.

This is a fork from arduino/ArduinoCore-samd on GitHub. This will be used to maintain Arduino support for SAM M0+ boards including the MattairTech MT-D21E and the MT-D11 (see <https://www.mattairtech.com/>). It adds support for new devices like the L21, C21, and D11. It also adds new clock sources, like a high speed crystal or internal oscillator.

This core is intended to be installed using Boards Manager (see below). To update from a previous version, click on MattairTech SAM M0+ Boards in Boards Manager, then click Update.

Differences from Arduino in Versioning: The MattairTech version number does not correspond to either the IDE version or to the upstream ArduinoCore-samd version. See the CHANGELOG for details on which upstream commits have been merged in to the MattairTech core.

What's New Beta (1.6.8-beta)

See Beta Builds section for installation instructions.

1.6.8-beta-b1:

- Fixed auto-reset not working on some versions of Windows
- Documentation updates

1.6.8-beta-b0:

- Added L21 and C21 support. Improved D11D and D11C support.
 - Use Tools->Microcontroller menu to select mcu.
- Both the core and bootloader have added support for:
 - external high-speed crystal (400KHz - 32MHz) using PLL
 - external 32.768KHz crystal using PLL
 - internal oscillator with USB calibration using DFLL
 - internal oscillator using DFLL in open-loop mode (or 48MHz RC oscillator with C21)
 - PLL_FRACTIONAL_ENABLED and PLL_FAST_STARTUP options
 - The clock source is selectable in the Tools->Clock Source menu
- New Tools->Serial Config menu for selecting different combinations of serial peripherals
- New Tools->Bootloader Size menu allows selection of bootloader size
- New Tools->USB Config menu simplifies USB configuration compared to previous core

- Updated variant.cpp table format for future CCL and GCLK use. See VARIANT_COMPLIANCE_CHANGELOG.
- Updated bootloader.
- Updated bossac upload tool (fixed support for SAML and SAMC)
- New CMSIS-Atmel package (this is different than from Arduino)
- Merged in all changes from upstream through SAMD CORE 1.6.14 (April 2017)

What's New Release (1.6.6)

This is out of date, use the beta for now.

- **1.6.6-mt3:**
 - Fixes compilation with CDC_UART and CDC_ONLY settings
- **1.6.6-mt2:**
 - Changes the default Communication setting to CDC_UART (from CDC_HID_UART)
- **1.6.6-mt1:**
 - New documentation section 'Special Notes'. Please read!
 - Updated ASCII pinouts to be more readable and less ambiguous.
 - Updated the Signed driver for Windows (extras directory) (see CHANGELOG for details)
 - Merged in changes from upstream (see CHANGELOG for details)
 - Fix warnings about deprecated recipe.ar.pattern
 - Merged in changes from upstream SAMD CORE 1.6.2 2015.11.03 (see CHANGELOG for details)

Features Summary

Feature	21J (64 pin)	21G (48 pin)	21E (32 pin)	D11 (24, 20, or 14 pin)
Board Variants	New board coming June, Generic 21J	Arduino Zero, Arduino M0, Generic 21G	MT-D21E, Generic 21E	MT-D11, Generic D11D14AM, Generic D11D14AS, Generic D11C14A
Processor	48 MHz 32-bit ARM Cortex M0+	48 MHz 32-bit ARM Cortex M0+	48 MHz 32-bit ARM Cortex M0+	48 MHz 32-bit ARM Cortex M0+
Flash Memory	Up to 256KB, L21 & C21 have RWW	Up to 256KB, L21 & C21 have RWW	Up to 256KB, L21 & C21 have RWW	16 KB (4KB used by bootloader)
SRAM	Up to 32KB (plus <=8KB LPSRAM on L21)	Up to 32KB (plus <=8KB LPSRAM on L21)	Up to 32KB (plus <=8KB LPSRAM on L21)	4 KB
Digital Pins	52 (51 for L21)	38 (37 for L21)	26 (25 for L21)	24-pin: 21, 20-pin: 17, 14-pin: 11
Analog Inputs	20 channels, 12-bit	14 channels, 12-bit	10 channels, 12-bit	24-pin: 10, 20-pin: 8, 14-pin: 5 (12-bit)
Analog Outputs	One 10-bit (two 12-bit on L21)	One 10-bit (two 12-bit on L21)	One 10-bit (two 12-bit on L21)	One 10-bit
PWM Outputs	18	14	14	8 (6 for 14-pin)
Interrupts	16	16	16	8 (7 for 14-pin)
USB	Full Speed Device and Host (not C21)	Full Speed Device and Host (not C21)	Full Speed Device and Host (not C21)	Full Speed Device
SERCOM	6	6	4	3 (2 for 14-pin)
UART (Serial)	Up to 3	Up to 3	Up to 2	Up to 2
SPI	Up to 3	Up to 2	1	1
I2C (WIRE)	Up to 3	Up to 2	1	1
I2S	D21 only	D21 only	D21 only	Not present
Voltage	1.62V-3.63V (2.7V-5.5V for the C21)	1.62V-3.63V (2.7V-5.5V for the C21)	1.62V-3.63V (2.7V-5.5V for the C21)	1.62V-3.63V
I/O Pin Current	D21: 7mA, L21: 5mA, C21: 6mA@5V	D21: 7mA, L21: 5mA, C21: 6mA@5V	D21: 7mA, L21: 5mA, C21: 6mA@5V	7 mA

Board Variants

Pin configuration and peripheral assignment information is now in the README.md for each board variant. README.md also now includes technical information on the new PinDescription table format.

- [MattairTech MT-D21E Rev B \(SAMx21Exxx\)](#)
- [MattairTech MT-D21E Rev A \(SAMD21ExxA\)](#)
- [MattairTech MT-D11 \(SAMD11D14AM\)](#)
- [MattairTech Generic D11C14A](#)
- MattairTech x21J based board (coming June)
- MattairTech Generic D11D14AS (coming soon)
- MattairTech Generic D11D14AM (coming soon)
- MattairTech Generic x21E (coming soon)
- MattairTech Generic x21G (coming soon)
- MattairTech Generic x21J (coming soon)
- [Arduino Zero \(arduino.cc\)](#)
- [Arduino M0 \(arduino.org\)](#)

Pin Configurations

MT-D21E rev B (D21E / L21E / C21E)

===== MattairTech MT-D21E rev B (ATSAMx21Exxx) =====												
Other	COM	PWM	Analog	INT	Arduino*			Arduino*	INT	PWM	COM	Other
=====												
XI32(+)					A0		RST					BOOT(+)
X032(+)					A1		Gnd					
DAC		*			2	A2	Vbat					
REFA		*			3	A3	A31	31	*			IO/B(+)
REFB		*		*	4	A4	A30	30	*			CLK(+)
DAC1(L)		*		*	5	A5	NC					
LED(+)	TCC10	*			6	A6	A28 (D/C)	28	*			
VM	TCC11	*			7	A7	A27	27	*			A/CS(+M)
	TCC00	*		NMI	8	A8	A23	23	*	TC41/TC01~	SS	
	TCC01	*		*	9	A9	A22	22	*	TC40/TC00~	MISO(+M)	
TX1	TCC02	*			10	A10	A19	19	*		SCK(+M)	
RX1	TCC03	*			11	A11	A18	18	*		MOSI(+M)	
TX2	TC30/TC40~			*	14	A14	A17	17	*	TCC21	SCL(+)	
RX2	TC31/TC41~				15	A15	A16	16	*	TCC20	SDA(+)	
						NC	NC					
M=Memory device installed						NC	NC		! Vcc is 3.3V by default.			
						Vbus	3.3V	DO NOT exceed 3.6V on Vcc or				
USB D- (D/L)(+), CAN TX (C)			TC50	24		A24	Vcc	any IO pin with the D21 or				
USB D+ (D/L)(+), CAN RX (C)			TC51	25		A25	Vin	L21 installed. 5V is allowed				
						Gnd	USB	Gnd	ONLY with the C21 installed.			

Chip Variant:												
D=D21, L=L21, C=C21												

* Most pins can be used for more than one function. The same port pin number printed on the board is also used in Arduino (without the 'A') for all of the supported functions (ie: digitalRead(), analogRead(), analogWrite(), attachInterrupt(), etc.).

+ This alternate function is enabled by default (+M functions enabled only when a memory device is installed). Thus, the associated header pin cannot be used. Solder jumpers can be used to enable or disable the alternate onboard function.

~ When two timers are shown, the second is for L21/C21. TC5 is TC1 on the L21/C21.

Silkscreen Legend:

Top: A circled pin means analog function and '*' means alternate function (see + above)

Bottom: A circled pin means analog function

Tools Menu Additions

Depending on the board variant, different menu options will appear in the Tools menu.

Microcontroller Menu

This menu will appear with boards that have multiple microcontroller options.

Clock Source Menu

There are up to four clock source choices, depending on board variant and microcontroller. They are:

- 32KHZ_CRYSTAL (default)
- HIGH_SPEED_CRYSTAL
- INTERNAL_OSCILLATOR
- INTERNAL_USB_CALIBRATED_OSCILLATOR

See Clock Source section for more information.

Bootloader Size Menu

With the D21, L21, and C21, the bootloader size can be configured as:

- 8KB_BOOTLOADER (default)
- 16KB_BOOTLOADER
- NO_BOOTLOADER

With the D11, the bootloader size can be configured as:

- 4KB_BOOTLOADER (default)
- NO_BOOTLOADER

Choose NO_BOOTLOADER if not using a bootloader (an external programmer will be used for sketch upload).

Serial Config Menu

This menu is used to select different combinations of serial peripherals. This is useful especially for the D11, which has a reduced pin count and number of SERCOMs. It can also be used to reduce FLASH and SRAM usage by selecting fewer UART peripherals, which are instantiated in the core, rather than only when including a library (like SPI and WIRE). Most board variants support two UART as an option.

USB Config Menu

This menu will appear with all microcontrollers except the C21, which does not have USB. The options are:

- CDC_ONLY (default)
- CDC_HID
- WITH_CDC

- HID_ONLY
- WITHOUT_CDC
- USB_DISABLED

Choose an option that best matches your code and library usage. Each option results in a different USB PID. Choose an option with CDC if you want auto-reset to function, or the serial monitor over USB. If CDC is not enabled, Serial will refer to Serial1 instead of SerialUSB. These options can be used to optimize FLASH and SRAM usage by allowing CDC to be disabled (or USB completely disabled).

Clock Source

There are up to four clock source choices, depending on board variant and microcontroller. They are:

- 32KHZ_CRYSTAL (default)
- HIGH_SPEED_CRYSTAL
- INTERNAL_OSCILLATOR
- INTERNAL_USB_CALIBRATED_OSCILLATOR

External 32.768KHz Crystal

The PLL will be used with the 32.768KHz crystal. PLL_FRACTIONAL_ENABLED can be defined, which will result in a more accurate 48MHz output frequency at the expense of increased jitter.

External High-Speed Crystal

HS_CRYSTAL_FREQUENCY_HERTZ must be defined with the external crystal frequency in Hertz. The crystal frequency must be between 400000Hz and 32000000Hz. The PLL will be used.

PLL_FRACTIONAL_ENABLED can be defined, which will result in a more accurate 48MHz output frequency at the expense of increased jitter. If PLL_FAST_STARTUP is defined, the crystal will be divided down to 1MHz - 2MHz, rather than 32KHz - 64KHz, before being multiplied by the PLL. This will result in a faster lock time for the PLL, however, it will also result in a less accurate PLL output frequency if the crystal is not divisible (without remainder) by 1MHz. In this case, define PLL_FRACTIONAL_ENABLED as well. By default, PLL_FAST_STARTUP is disabled.

PLL_FAST_STARTUP is also useful for USB host mode applications. See datasheet USB electrical characteristics. The crystal frequency must be at least 1000000Hz when PLL_FAST_STARTUP is defined.

Internal Oscillator

The DFLL will be used in open-loop mode, except with the C21 which lacks a DFLL, so the internal 48MHz RC oscillator is used instead. NVM_SW_CALIB_DFLL48M_FINE_VAL is the fine calibration value for DFLL open-loop mode. The coarse calibration value is loaded from NVM OTP (factory calibration values).

Internal Oscillator with USB Calibration

This is available for the D21, D11, or L21. It will also use the DFLL in open-loop mode, except when

connected to a USB port with data lines (and not suspended), then it will calibrate against the USB SOF signal. NVM_SW_CALIB_DFLL48M_FINE_VAL is the fine calibration value for DFLL open-loop mode. The coarse calibration value is loaded from NVM OTP (factory calibration values).

Clock Generators

0. MAIN (mcu)
1. XOSC (high speed crystal)
2. OSCULP32K (initialized at reset for WDT on D21 and D11)
3. OSC_HS (the reset default internal RC oscillator is put here at 8MHz, except with C21)

Analog Reference

TODO: more info

- **D21 / D11**

- AR_INTERNAL1V0
- AR_INTERNAL_INTVCC0
- AR_INTERNAL_INTVCC1
- AR_EXTERNAL_REFA
- AR_EXTERNAL_REFB
- AR_DEFAULT (this also uses 1/2 gain on each input)

- **L21**

- AR_INTREF
- AR_INTERNAL_INTVCC0
- AR_INTERNAL_INTVCC1
- AR_EXTERNAL_REFA
- AR_EXTERNAL_REFB
- AR_INTERNAL_INTVCC2
- AR_INTREF_1V0
- AR_INTREF_1V1
- AR_INTREF_1V2
- AR_INTREF_1V25
- AR_INTREF_2V0
- AR_INTREF_2V2
- AR_INTREF_2V4
- AR_INTREF_2V5
- AR_DEFAULT = AR_INTERNAL_INTVCC2
- AR_INTERNAL1V0 = AR_INTREF (Default INTREF for SAML is 1.0V)

- **C21**

- AR_INTREF
- AR_INTERNAL_INTVCC0
- AR_INTERNAL_INTVCC1
- AR_EXTERNAL_REFA

- AR_EXTERNAL_DAC
- AR_INTERNAL_INTVCC2
- AR_INTREF_1V024
- AR_INTREF_2V048
- AR_INTREF_4V096
- AR_DEFAULT = AR_INTERNAL_INTVCC2
- AR_INTERNAL1V0 = AR_INTREF (Default INTREF for SAMC is 1.024V)
- **Common**
 - AR_INTERNAL = AR_INTERNAL_INTVCC0
 - AR_INTERNAL2V23 = AR_INTERNAL_INTVCC0 (2.23V only when Vcc = 3.3V)
 - AR_INTERNAL1V65 = AR_INTERNAL_INTVCC1 (1.65V only when Vcc = 3.3V)
 - AR_EXTERNAL = AR_EXTERNAL_REFA

Chip Specific Notes

SAMD21

- When USB is disabled, pullups will be enabled on PA24 and PA25 to avoid excessive current consumption (<1mA) due to floating pins. Note that it is not necessary to enable pull resistors on any other pins that are floating. Errata: Disable pull resistors on PA24 and PA25 manually before switching to a peripheral.

SAML21

- There are two DACs, DAC0 and DAC1. Both are supported. Because changing the configuration of one DAC requires disabling both, there will be about a 40us period when the second DAC is disabled. Most of this time is due to an errata that requires a delay of at least 30us when turning off the DAC while refresh is on. The L21 DACs have a refresh setting which must be enabled in this core.
- The analog reference has additional options on the L21 and C21. See Analog Reference section.
- On the L21, SERCOM5 is in a low power domain. The Fm+ and HS modes of I2C (wire) are not supported.
- The SAML and SAMC have double-buffered TCs, which are supported in the core.
- The CHANGE and RISING interrupt modes on pin A31 do not seem to work properly on the L21.
- The L21 has two performance levels that affect power consumption. During powerup, the L21 starts at the lowest performance level (PL0). The startup code changes to the highest performance level (PL2) in order to support 48MHz and USB (among other things).
- Two Flash Wait States are inserted for the L21 and C21 (the D21/D11 use one wait state).

SAMC21

- There are two SAR ADCs. Both are supported. The PinDescription table determines the peripheral instance and pin mapping.

- The analog reference has additional options on the L21 and C21. See Analog Reference section.
- The SAML and SAMC have double-buffered TCs, which are supported in the core.
- Two Flash Wait States are inserted for the L21 and C21 (the D21/D11 use one wait state).
- The C21 requires internal pull resistors to be activated on floating pins to minimize power consumption (not needed on D21/D11 or L21).
- The C21 uses the minimum sampling time so that rail-to-rail and offset compensation works. Offset compensation adds 3 ADC clock cycles, so the total is 4 clock cycles. The D21, D11, and L21 use the maximum sampling time.

SAMD11

- The D11D has three SERCOM. The D11C has two sercom (no sercom2).
- When USB is disabled, pullups will be enabled on PA24 and PA25 to avoid excessive current consumption (<1mA) due to floating pins. Note that it is not necessary to enable pull resistors on any other pins that are floating. Errata: Disable pull resistors on PA24 and PA25 manually before switching to a peripheral.

Reducing SRAM/FLASH Usage on the D11

TODO

Differences Between MattairTech and Arduino Cores

- TODO
- Table summarizing which core files are modified and by how much
- Changes due to adding/changing features vs porting to new chip

Random Notes (TODO)

- TONE: TC5 does not exist on the D11. Using TC2 instead (TC1 on the D11C14 as TC2 is not routed to pins). It will conflict with the 2 associated TC analogWrite() pins.
- D21: Enables wakeup capability on pin in case being used during sleep (WAKEUP always enabled on SAML and SAMC)
- All pins (digital and analog) setup in STARTUP mode (enable INEN and set default pull direction to pullup (pullup will not be enabled))
- INEN enabled for both input and output (but not analog)
- pinPeripheral now handles disabling the DAC (if active). Note that on the L21, the DAC output would interfere with other peripherals if left enabled, even if the analog peripheral is not selected.
- Pull resistors enabled only if pin attributes allow and only if pin is not configured as output.
- Pull direction (pullup or pulldown) is now set with pinMode only (defaults to pullup if pinMode never called).

Serial Monitor

To print to the Serial Monitor over USB, use 'Serial'. Serial points to SerialUSB (Serial1 and Serial2 are UARTs). Unlike most Arduino boards (ie. Uno), SAMD boards do not automatically reset when the serial monitor is opened. To see what your sketch outputs to the serial monitor from the beginning, the sketch must wait for the SerialUSB port to open first. Add the following to setup():

```
while (!Serial) ;
```

Remember that if the sketch needs to run without SerialUSB connected, another approach must be used. You can also reset the board manually with the Reset button if you wish to restart your sketch. However, pressing the Reset button will reset the SAMD chip, which in turn will reset USB communication. This interruption means that if the serial monitor is open, it will be necessary to close and re-open it to restart communication.

When USB CDC is not enabled, Serial will instead refer to Serial1, which is the first UART.

Code Size and RAM Usage (1.6.5-mt2)

TODO: Update this. Maybe just for D11 and move to D11 Chip Specific Notes.

Sketch and Configuration	MT-D21E (Code + RAM)	MT-D11 (Code + RAM)
Blink (CDC + HID + UART)	7564 + 1524	7452 + 1424
Blink (CDC + UART)	6588 + 1496	6484 + 1396
Blink (CDC Only)	5248 + 1304	5192 + 1300
Blink (UART Only)	3828 + 336	3716 + 236
Blink (No USB or UART)	2472 + 144	2416 + 140
Datalogger (No USB or UART)	10340 + 948	10260 + 944

- 180 bytes of flash can be saved on the MT-D11 by using PIN_MAP_COMPACT (see 'New PinDescription Table' below).
- Datalogger compiled without USB or UART support, but with SPI and SD (with FAT filesystem) support. Serial output was disabled.
- Note that USB CDC is required for auto-reset into the bootloader to work (otherwise, manually press reset twice in quick succession).
- USB uses primarily 3 buffers totaling 1024 bytes. The UART uses a 96 byte buffer. The

banzai() function (used for auto-reset) resides in RAM and uses 72 bytes.

- Any combination of CDC, HID, or UART can be used (or no combination), by using the Tools->Communication menu.

Detailed Memory Usage Output After Compilation

The flash used message at the end of compilation is not correct. The number shown represents the .text segment only. However, Flash usage = .text + .data segments (RAM usage = .data + .bss segments). In this release, two programs are run at the end of compilation to provide more detailed memory usage. To enable this output, go to File->Preferences and beside "Show verbose output during:", check "compilation".

Just above the normal flash usage message, is the output from the size utility. However, this output is also incorrect, as it shows .text+.data in the .text field, but 0 in the .data field. However, the .text field does show the total flash used. The .data field can be determined by subtracting the value from the normal flash usage message (.text) from the value in the .text field (.text+.data). The .bss field is correct.

Above the size utility output is the output from the nm utility. The values on the left are in bytes. The letters stand for: T(t)=.text, D(d)=.data, B(b)=.bss, and everything else (ie: W) resides in flash (in most cases).

Installation

Driver Installation

Windows

Prior to core version 1.6.6-mt1, sketches compiled with both CDC and HID USB code by default, thus requiring a CDC driver for the bootloader and a CDC-HID driver for sketches. Now that PluggableUSB is supported, sketches compile with only CDC code by default. Thus, only one driver is needed. Since HID and MIDI are currently supported (and MSD potentially in the future), driver installation will be required for each different combination of USB devices. There are currently four USB composite device combinations that include CDC as well as a CDC only device. Each supported combination has a unique USB VID:PID pair, and these are listed in the .inf file. Once the first device is installed (the CDC only device), future installations *might* be automatic, otherwise, you may direct the installer to the same .inf file. The drivers are signed and support both 32 and 64 bit versions of Windows XP(SP3), Vista, 7, 8, and 10. Note that the Windows 10 generic CDC drivers work as well.

1. If you do not already have the SAM-BA bootloader installed, see below.
2. Download https://www.mattairtech.com/software/MattairTech_CDC_Driver_Signed.zip and unzip into any folder. Note that the Windows 10 generic CDC drivers work as well.
3. Plug in the board. The LED should fade when the bootloader is running (or blink if the test sketch is running).
4. Windows will detect the board. Point the installer to the folder from above to install the bootloader driver.
5. If you don't intend on using Arduino, you can skip the rest of this list. See Using Bossac Standalone below.
6. If you do not already have the test firmware installed (comes preinstalled), see Using Bossac Standalone below.
7. Press the reset button to run the test firmware (if needed). The LED will blink.
8. Windows will detect the board. Point the installer to the above folder to install the sketch driver (if needed).
9. Continue with SAM M0+ Core Installation below.

Linux

0. No driver installation is needed.
1. On some distros, you may need to add your user to the same group as the port (ie: dialout) or set udev rules (See the file <https://github.com/mattairtech/ArduinoCore-samd/tree/master/drivers/99-mattairtech-USB-CDC.rules>).
2. You MAY have to install and use Arduino as the root user in order to get reliable access to the serial port.
 - This is true even when group permissions are set correctly, and it may fail after previously working.
 - You can also create/modify a udev rule to set permissions on the port so *everyone* can

read / write.

3. If you are running modemmanager (ie: Ubuntu), disable it, or use the udev rules file above.
4. Continue with SAM M0+ Core Installation below.

OS X

1. OS X support currently in beta (see below), the following instructions are only for 1.6.6-mtX and below.
2. Only the 256 KB chip variants work with the OS X version of the upload tool, bossac.
3. First, you will need to open boards.txt and change mattairtech_mt_d21e_bl8k.upload.tool to equal arduino:bossac.
4. Open platform.txt and change tools.bossac.path to equal{runtime.tools.bossac-1.6.1-arduino.path}.
5. No driver installation is needed.
6. Plug in the board. You may get a dialog box asking if you wish to open the "Network Preferences":
 - Click the "Network Preferences..." button, then click "Apply".
 - The board will show up as "Not Configured", but it will work fine.
7. Continue with SAM M0+ Core Installation below.

SAM M0+ Core Installation

- To update from a previous version, click on MattairTech SAM M0+ Boards in Boards Manager, then click Update.
1. The MattairTech SAM M0+ Core requires Arduino 1.6.7 or above (including 1.8.x).
 2. In the Arduino IDE, click File->Preferences.
 3. Click the button next to Additional Boards Manager URLs.
 4. Add https://www.mattairtech.com/software/arduino/package_MattairTech_index.json.
 5. Save preferences, then open the Boards Manager.
 6. Install the Arduino SAM M0+ Boards package. Use version 1.6.2 or higher.
 7. Install the MattairTech SAM M0+ Boards package.
 8. Close Boards Manager, then click Tools->Board->MattairTech MT-D21E (or MT-D11).
 9. Select the MCU with the now visible Tools->Microcontroller menu (if present).
 10. If you do not already have the bootloader or blink sketch installed, see SAM-BA USB CDC Bootloader below.
 11. Plug in the board. The blink sketch should be running.
 12. Click Tools->Port and choose the COM port. Note that the board indicated may not match the chosen board*
 13. You can now upload your own sketch.

** Currently, with MattairTech boards, USB PIDs are shared across boards (but they are different based on Tools->USB Config). This will result in Tools->Port showing "MattairTech MT-D21E (rev B)" for all MattairTech boards.*

Uploading the First Sketch

1. In the Arduino IDE (1.6.7 or above), open File->Examples->01.Basics->Blink.
2. Change the three instances of '13' to 'LED_BUILTIN'.
3. Be sure the correct options are selected in the Tools menu (see AVR Core Installation above).
4. With the board plugged in, select the correct port from Tools->Port.
5. Click the Upload button. After compiling, the sketch should be transferred to the board.
6. Once the bootloader exits, the blink sketch should be running.

Beta Builds

Periodically, a beta is released for testing.

The beta builds are available through Boards Manager. If you want to install them:

1. Open the **Preferences** of the Arduino IDE.
2. Add this URL
`https://www.mattairtech.com/software/arduino/beta/package_MattairTech_index.json` in the **Additional Boards Manager URLs** field, and click OK.
3. Open the **Boards Manager** (menu Tools->Board->Board Manager...)
4. Install **MattairTech SAM M0+ Boards - Beta build**
5. Select one of the boards under **MattairTech SAM M0+ Beta Build XX** in Tools->Board menu
6. Compile/Upload as usual

The Arduino IDE will notify the user if an update to the beta is available, which can then be installed automatically. Alternatively, if a particular beta is needed, replace the url in step 2 with:

`https://www.mattairtech.com/software/arduino/beta/package_MattairTech_sam_m0p-${VERSION}-beta-b${BUILD_NUMBER}_index.json` where `${VERSION}` and `${BUILD_NUMBER}` match the beta name as shown in the CHANGELOG (ie: `package_MattairTech_sam_m0p-1.6.7-beta-b0_index.json`). In this case, the IDE will not notify the user of updates.

New PinDescription Table

Technical information on the new PinDescription table format is now in the README.md that accompanies each board variant. See board variants above.

Note that a new column (GCLKCCL) was added for 1.6.8-beta-b0.

MATTAIRTECH_ARDUINO_SAMD_VARIANT_COMPLIANCE in variant.h is used to track versions. If using board variant files with the old format, the new core will still read the table the old way, losing any new features introduced by the new column. Additionally, new definitions have been added for L21 and C21 support.

Each pin can have multiple functions.

The PinDescription table describes how each of the pins can be used by the Arduino core. Each pin can have multiple functions (ie: ADC input, digital output, PWM, communications, etc.), and the PinDescription table configures which functions can be used for each pin. This table is mainly accessed by the pinPeripheral function in wiring_private.c, which is used to attach a pin to a particular peripheral function. The communications drivers (ie: SPI, I2C, and UART), analogRead(), analogWrite(), analogReference(), attachInterrupt(), and pinMode() all call pinPeripheral() to verify that the pin can perform the function requested, and to configure the pin for that function. Most of the contents of pinMode() are now in pinPeripheral().

Pin Mapping

There are different ways that pins can be mapped. Typically, there is no relation between the arduino pin number used, and the actual port pin designator. Thus, the pcb must be printed with the arduino numbering, otherwise, if the port pin is printed, a cross reference table is needed to find the arduino pin number. However, this results in the least amount of space used by the table. Another method, used by default by the MT-D21E and MT-D11, maps Arduino pin numbers to the actual port pin number (ie: Arduino pin 28 = Port A28). This works well when there is only one port (or if the PORTB pins are used for onboard functions and not broken out). PIO_NOT_A_PIN entries must be added for pins that are used for other purposes or for pins that do not exist (especially the D11), so some FLASH space may be wasted. For an example of both types, see variant.cpp from the MT-D11 variant.

See Board Variants above for more technical information on the PinDescription table.

See [WVariant.h](#) for the definitions used in the table.

Possible Future Additions/Changes

- Timer library is currently under development (like TimerOne, plus input capture, plus ??)
- OS X support currently in beta testing
- Reduce SRAM usage by USB endpoint buffers by only allocating endpoints actually used (D11 especially)
- Drivers for MT-D21E optional memory devices (SRAM, FLASH, EEPROM)
- USB Host mode CDC ACM (partially complete; BSD-like license?)
- Features for lower power consumption (library?)
- Reliability and security enhancements
- Enhanced SD card library
- Optional use of single on-board LED as USB activity LED
- MSC (Mass Storage) USB Device Class
- Polyphonic tone
- Wired-AND, Wired-OR for port pins

- High-speed port pin access (IOBUS)
- Libraries for some hardware I plan on using:
 - ◆ TFT LCD (CFAF128128B-0145T)
 - ◆ Motor controller
 - ◆ IR decoder
 - ◆ I2S DAC/AMP and I2S MEMS microphone
 - ◆ Battery management IC
 - ◆ XBee/Xbee Pro devices
 - ◆ RS485
 - ◆ Several I2C (Wire) sensor devices:
 - ◆ Accelerometer/magnetometer (LSM303CTR)
 - ◆ Barometer/altimeter (LPS22HBTR)
 - ◆ Humidity/temperature
 - ◆ Light/color sensor

ChangeLog

The Changelog has moved to a separate file named CHANGELOG. The most recent changes are still in the 'What's New' section above.

Troubleshooting

- **Tools->Port shows wrong board**
 - Currently, with MattairTech boards, USB PIDs are shared across boards (but they are different based on Tools->USB Config). This will result in Tools->Port showing "MattairTech MT-D21E (rev B)" for all MattairTech boards.
- **Tools->USB Config menu**
 - Currently, the Tools->USB Config menu (was Tools->Communications) must be used to select the communications configuration. This configuration must match the included libraries. For example, when including the HID and Keyboard libraries, you must select an option that includes HID (all options except CDC_ONLY or USB_DISABLED). This menu is currently needed to select the USB PID that matches the USB device configuration (needed for some versions of Windows). It is also used to control if CDC support is compiled (CDC is always enabled in the stock Arduino core). Auto reset requires CDC to be enabled.
 - Be sure that the Tools->Communications menu matches the sketch and libraries you are compiling.
 - Different combinations of USB devices will result in different COM port assignments in Windows.

- **Include platform specific libraries**
 - You may need to manually include platform specific libraries such as SPI.h, Wire.h, and HID.h.
- **Errors when compiling, uploading, or burning the bootloader**
 - Be sure to install the Arduino samd core before installing the MattairTech sam m0+ core. If you have problems upgrading the IDE to 1.6.6, you may need to uninstall both the Arduino and MattairTech cores, then re-install in the proper order. Use Arduino core 1.6.2 or above.
- **On Linux, disable modem manager (Ubuntu)**
- **Do not perform a manual auto-reset** (using a terminal program to change baud to 1200)
- **Boards Manager must be opened twice to see some updates** (only applies to some old IDE versions)

SAM-BA USB CDC Bootloader (Arduino Compatible)

The SAM-BA bootloader has both a CDC USB interface, and a UART interface. It is compatible with the Arduino IDE, or it can be used with the Bossac tool standalone. With Arduino, auto-reset is supported (automatically runs the bootloader while the sketch is running) as well as automatic return from reset. The SAM-BA bootloader described here adds to the Arduino version, which in turn is based on the bootloader from Atmel. The Arduino version added several features, including three new commands (Arduino Extended Capabilities) that increase upload speed. The bootloader normally requires 8 KB FLASH, however, a 4 KB version can be used for the D11 chips.

Bossac is a command line utility for uploading firmware to SAM-BA bootloaders. It runs on Windows, Linux, and OS X. It is used by Arduino to upload firmware to SAM and SAM M0+ boards. The version described here adds to the Arduino version (<https://github.com/shumatech/BOSSA>, Arduino branch), which in turn is a fork from the original Bossa (<http://www.shumatech.com/web/products/bossa>). It adds support for more SAM M0+ chips (D21, L21, C21, and D11).

Note that only the Arduino or Mattairtech versions of bossac are currently supported for SAM M0+ chips. Neither the stock bossac (or Bossa) nor the Atmel SAM-BA upload tool will work.

Arduino Extended Capabilities:

- X: Erase the flash memory starting from ADDR to the end of flash.
- Y: Write the content of a buffer in SRAM into flash memory.
- Z: Calculate the CRC for a given area of memory.

The bootloader can be started by:

- Tapping reset twice in quick succession (BOOT_DOUBLE_TAP).
- Holding down button A (BOOT_LOAD_PIN) while powering up.
- Clicking 'Upload Sketch' in the Arduino IDE, which will automatically start the bootloader (when CDC is enabled).
- If the application (sketch) area is blank, the bootloader will run.

Otherwise, it jumps to application and starts execution from there. The LED will light during bootloader execution. Note that the 4KB bootloader does not support the Arduino Extended Capabilities. However, BOOT_DOUBLE_TAP does fit into the SAMD11 4KB bootloader.

When the Arduino IDE initiates the bootloader, the following procedure is used:

1. The IDE opens and closes the USB serial port at a baud rate of 1200bps. This triggers a “soft erase” procedure.
2. The first row of application section flash memory is erased by the MCU. If it is interrupted for any reason, the erase procedure will likely fail.
3. The board is reset. The bootloader (which always runs first) detects the blank flash row, so bootloader operation resumes.
4. Opening and closing the port at a baud rate other than 1200bps will not erase or reset the SAM M0+.

See [bootloaders/zero/README.md](https://github.com/shumatech/BOSSA/blob/master/README.md) for more technical information on the bootloader.

Bootloader Firmware Installation

Bootloader Installation Using the Arduino IDE

1. If you do not already have the MattairTech SAM M0+ core installed, see SAM M0+ Core Installation above.
2. Plug in the SAM M0+ board. The bootloader must be running to (press reset twice within 500ms).
3. Plug an Atmel ICE into USB, then connect it to the powered SAM M0+ board. A green LED should light on the Atmel ICE.
4. Click Tools->Programmer->Atmel ICE.
5. Click Tools->Board->MattairTech MT-D21E (or whichever board you are using).
6. Click Tools->Microcontroller and select your MCU (if menu present).
7. Click Tools->Burn Bootloader. Ignore any messages about not supporting shutdown or reset.
8. Continue with driver installation above.

A running sketch may interfere with the bootloader installation process. Be sure you are running the existing bootloader or using a blank chip.

Bootloader Installation Using Another Tool (ie: Atmel Studio, openocd)

1. Download the bootloader from <https://www.mattairtech.com/software/arduino/SAM-BA-bootloaders-zero-mattairtech.zip>.
2. Unzip to any directory. Be sure that a bootloader is available for your particular MCU.
3. Follow the procedures for your upload tool to upload the firmware.
 - Perform a chip erase first. Be sure no BOOTPROT bits are set.
 - Install the binary file to 0x00000000 of the FLASH.
 - You can optionally set the BOOTPROT bits to 8KB (or 4KB for the MT-D11). The Arduino installation method does not set these.
 - You can optionally set the EEPROM bits or anything else. The Arduino installation method uses factory defaults.
4. Continue with driver installation above.

Bootloader Binaries

The bootloaders/zero/binaries directory contains all of the SAM-BAm0+ bootloaders built by the build_all_bootloaders.sh script.

MattairTech Boards

MattairTech boards are all configured with only one interface: SAM_BA_USBCDC_ONLY (except C21, which uses SAM_BA_UART_ONLY). CLOCKCONFIG_CLOCK_SOURCE is set to CLOCKCONFIG_INTERNAL_USB (CLOCKCONFIG_INTERNAL for the C21). Only the main LED is

defined. BOOT_LOAD_PIN is not defined, but BOOT_DOUBLE_TAP_ENABLED is.

Arduino/Genuino Boards

Arduino/Genuino boards are all configured with both interfaces. CLOCKCONFIG_CLOCK_SOURCE is set to CLOCKCONFIG_32768HZ_CRYSTAL. All LEDs that are installed for each board are defined (and some have LED_POLARITY_LOW_ON set). BOOT_LOAD_PIN is not defined, but BOOT_DOUBLE_TAP_ENABLED is.

Generic Boards

The generic boards are all configured to minimize external hardware requirements. Only one interface is enabled: SAM_BA_USBCDC_ONLY (except C21, which uses SAM_BA_UART_ONLY). CLOCKCONFIG_CLOCK_SOURCE is set to CLOCKCONFIG_INTERNAL_USB (CLOCKCONFIG_INTERNAL for the C21), so no crystal is required. No LEDs are defined. BOOT_LOAD_PIN is not defined, but BOOT_DOUBLE_TAP_ENABLED is, since it uses the reset pin.

Using Bossac Standalone

TODO: Update <https://www.mattairtech.com/software/SAM-BA-bootloader-test-firmware.zip> with new chips (L21 and C21).

When using Bossac standalone, you will need to ensure that your application starts at 0x00002000 for 8 KB bootloaders, and 0x00001000 for 4 KB bootloaders. This is because the bootloader resides at 0x00000000. This can be accomplished by passing the following flag to the linker (typically LDFLAGS in your makefile; adjust for your bootloader size):

```
-Wl,--section-start=.text=0x2000
```

You can also use a linker script. See the MattairTech SAM M0+ package for examples. Be sure to generate and use a binary file. Many makefiles are set up to generate an elf, hex, and bin already.

Download Bossac from:

- <https://www.mattairtech.com/software/arduino/bossac-1.7.0-mattairtech-1-mingw32.tar.gz> (Windows 32 bit and 64 bit)
- https://www.mattairtech.com/software/arduino/bossac-1.7.0-mattairtech-1-x86_64-linux-gnu.tar.gz (Linux 64 bit)
- <https://www.mattairtech.com/software/arduino/bossac-1.7.0-mattairtech-1-i686-linux-gnu.tar.gz> (Linux 32 bit)
- https://www.mattairtech.com/software/arduino/bossac-1.7.0-mattairtech-1-x86_64-apple-darwin.tar.gz (OS X 64 bit)

Linux 64 bit users can also download Bossa (GUI) and bossash (shell) from:

- https://www.mattairtech.com/software/arduino/Bossa-1.7.0-mattairtech-1-x86_64-linux-gnu.tar.gz (Linux 64 bit)

As an example, bossac will be used to upload the test firmware (blink sketch):

1. Download firmware from <https://www.mattairtech.com/software/SAM-BA-bootloader-test-firmware.zip> and unzip.
2. If you have not already installed the bootloader driver, see Driver Installation above.
3. Be sure there is a binary that matches your chip. On the command line (change the binary to match yours):

```
bossac.exe -d --port=COM5 -U true -i -e -w -v Blink_Demo_ATSAM21E18A.bin -R
```

4. On Linux --port might be /dev/ttyACM0. If the device is not found, remove the --port argument for auto-detection.
5. See <http://manpages.ubuntu.com/manpages/vivid/man1/bossac.1.html> for details.
6. The board should reset automatically and the sketch should be running.

USB Mass Storage Bootloader

Source code and binaries available at <https://github.com/mattairtech/SAMD-MSD-Bootloader>.

A USB Mass Storage Class device (MSC or MSD) bootloader can be optionally installed. This will allow programming of the FLASH without an external programmer. Additionally, no special software is required on the host computer. The bootloader occupies the first 16KB of FLASH, leaving the rest for the user firmware. The BOOTPROT fuse bits (2:0) can be set 0x01, which will protect the first 16KB of FLASH from internal or external programming (from 0x00000000 to 0x00004000). Note that the MSD bootloader does not use an external crystal, as it uses USB clock recovery (DFLL tuned using the USB SOF signal).

Installation

The Mass Storage Bootloader can be installed using atprogram.exe (included with Atmel Studio) with the following commands (ATSAMD21E18A shown):

```
atprogram -t atmelice -i SWD -d atsamd21e18a -cl 500khz program -c --verify -f  
c:\msd_bootloader_256_flash.hex
```

The three BOOTPROT fuse bits (2:0) can be set to 0x01 (16KB):

```
atprogram -t atmelice -i SWD -d atsamd21e18a -cl 500khz write -fs -o 0x00804000 --values f9
```

The blink program (compiled with an offset of 0x00004000) can then be installed using the Mass Storage Bootloader.

Special Requirements when Compiling Software

- Because the user firmware will begin executing at FLASH byte address 0x00004000, you must pass the following flag to the linker (typically LDFLAGS in your makefile):

```
-Wl,--section-start=.text=0x4000
```

- If using the MattairTech SAM M0+ Arduino core, be sure to select 16KB bootloader size in the Tools->Bootloader Size menu.
- Be sure to generate a binary file. Most makefiles are set up to generate an elf, hex, and bin already. You will need the bin file.
- You will need to rename the binary file to FLASH.BIN.

Entering the bootloader and programming the firmware

- Enter the bootloader by pressing button A while powering up the board from USB. Or, hold button A while pressing and releasing button B (if configured as RST). Button A must be connected to pin A27 via solder jumper J13 (this will already be soldered if you ordered the bootloader option). Note that when no user firmware is installed, the bootloader will not

automatically run, so you must always use the bootloader button. When the bootloader is run for the first time, the host operating system may take a small amount of time to install drivers. Drivers are already included with the OS, so there is nothing more to download. Once loaded, the LED will begin blinking at 2Hz.

- Mount the “FLASH disk” if it is not mounted automatically. The only file on the entire volume will be FLASH.BIN. This file represents the entire FLASH contents and will always exist. The file date will always be the same upon mounting (2/14/1989). You can read this file simply by copying it to your hard drive. It will include the installed firmware plus 0xFF for the remainder of the file (up to the end of the FLASH).
- Program the FLASH by copying your new FLASH.BIN over the existing copy on the “FLASH disk”. On Windows, you can do this with a file manager. On OS-X (and possibly Linux), you will need to use the cp command, which should already be present. Open up a console (Terminal on OS-X) and type (adjust for your system):

```
cp FLASH.BIN '/run/media/cygnus/MT-D21E MSD'
```

- Be sure to unmount the volume before running your new firmware, so that any disk caches are flushed.
- To run your firmware, simply reset or cycle power without pressing button A.
- Technical notes: The startup portion of the bootloader will run prior to executing your firmware. This startup code will enable the button A pullup resistor, wait 8ms for the debouncing capacitor to charge, then test the state of the button. If it is not pressed, the user firmware will be executed as follows:
 - The stack pointer location will be rebased to 0x00004000
 - The interrupt vector table will be rebased to (0x00004000 & SCB_VTOR_TBLOFF_Msk)
 - A jump will be performed to the user firmware reset vector.

Blink Demo

This currently describes Rev A boards only. Rev B boards will have a test program soon.

A demo program comes pre-installed. It simply blinks the LED at 1Hz using an internal clock source. The hex files can be found on the MT-D21E product page at <https://www.mattairtech.com/>. The blink demo was compiled using the Atmel Standalone Toolchain for Linux. It makes use of Atmel Software Framework (ASF) so it is rather large for a blink program. I can send the source upon request. I will post source if I ever recompile a simpler version that does not depend on ASF.

SAM-BA (USB CDC) Bootloader

Both the SAM-BA USB CDC bootloader and the blink sketch were pre-installed by using the Arduino IDE. Neither the region lock bits nor the security bit is set. The fuses are left at default settings.

Without Bootloader

The Blink program can be installed using atprogram.exe (included with Atmel Studio) with the following commands (ATSAMD21E18A shown):

```
atprogram -t atmelice -i SWD -d atsamd21e18a -cl 500khz program -c --verify -f  
c:\MT_D21E_Blink_256_no_offset_flash.hex
```

Troubleshooting / FAQ

- Released rev B on March 15, 2017
- Linux users: disable modemmanager (Ubuntu), install udev rules (see driver installation)

Support Information

Please check the MattairTech website (<http://www.MattairTech.com/>) for firmware and software updates. Email me if you have any feature requests, suggestions, or if you have found a bug. If you need support, please contact me (email is best). You can also find support information at the MattairTech website. A support forum is planned. Support for Atmel ARM in general can be found at <http://www.at91.com/>.

Justin Mattair
MattairTech LLC
PO Box 1079
Heppner, OR 97836 USA
justin@mattair.net (preferred)
541-626-1531 (message only)
<http://www.mattairtech.com/>

Legal

Copyright / Licenses

Arduino core files / SAM-BA Bootloader:

This core has been developed by Arduino LLC in collaboration with Atmel.
This fork developed by Justin Mattair of MattairTech LLC.

Copyright (c) 2017 Arduino LLC. All right reserved.
Copyright (c) 2017 MattairTech LLC. All rights reserved.

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

MSD Bootloader:

Portions of this code are copyright (c) 2009-2015 Justin Mattair (www.mattairtech.com)

Portions of this code are copyright © 2003-2014, Atmel Corporation (<http://www.atmel.com/>):

```
/**
 * \file
 *
 * \brief User Interface
 *
 * Copyright (c) 2014 Atmel Corporation. All rights reserved.
 *
 * \asf_license_start
 *
 * \page License
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions are met:
 *
 * 1. Redistributions of source code must retain the above copyright notice,
 *    this list of conditions and the following disclaimer.
 *
 * 2. Redistributions in binary form must reproduce the above copyright notice,
 *    this list of conditions and the following disclaimer in the documentation
 *    and/or other materials provided with the distribution.
 *
 * 3. The name of Atmel may not be used to endorse or promote products derived
```



```
* from this software without specific prior written permission.
*
* 4. This software may only be redistributed and used in connection with an
* Atmel microcontroller product.
*
* THIS SOFTWARE IS PROVIDED BY ATMEL "AS IS" AND ANY EXPRESS OR IMPLIED
* WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
* MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT ARE
* EXPRESSLY AND SPECIFICALLY DISCLAIMED. IN NO EVENT SHALL ATMEL BE LIABLE FOR
* ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
* STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN
* ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
* POSSIBILITY OF SUCH DAMAGE.
*
* \asf_license_stop
*
*/
```

Portions of this code are Copyright (C) 2009-2012 ARM Limited. All rights reserved.

```
* @note
* Copyright (C) 2009-2012 ARM Limited. All rights reserved.
*
* @par
* ARM Limited (ARM) is supplying this software for use with Cortex-M
* processor based microcontrollers. This file can be freely distributed
* within development tools that are supporting such ARM based processors.
*
* @par
* THIS SOFTWARE IS PROVIDED "AS IS". NO WARRANTIES, WHETHER EXPRESS, IMPLIED
* OR STATUTORY, INCLUDING, BUT NOT LIMITED TO, IMPLIED WARRANTIES OF
* MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE APPLY TO THIS SOFTWARE.
* ARM SHALL NOT, IN ANY CIRCUMSTANCES, BE LIABLE FOR SPECIAL, INCIDENTAL, OR
* CONSEQUENTIAL DAMAGES, FOR ANY REASON WHATSOEVER.
```

Portions of this code are copyright © 2003-2014, Dean Camera (www.fourwalledcubicle.com)
Specifically, the virtual FAT implementation from his MSD bootloader is used in the MT-D21E bootloader:

```
/*
    LUFA Library
    Copyright (C) Dean Camera, 2014.

    dean [at] fourwalledcubicle [dot] com
    www.lufa-lib.org
*/

/*
    Copyright 2014 Dean Camera (dean [at] fourwalledcubicle [dot] com)
```

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that the copyright notice and this permission notice and warranty disclaimer appear in supporting documentation, and that the name of the author not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

The author disclaims all warranties with regard to this software, including all implied warranties of merchantability and fitness. In no event shall the author be liable for any

special, indirect or consequential damages or any damages whatsoever resulting from loss of use, data or profits, whether in an action of contract, negligence or other tortious action, arising out of or in connection with the use or performance of this software.

*/

Portions of this code are Copyright (C) 2009-2012 ARM Limited. All rights reserved.

* @note

* Copyright (C) 2009-2012 ARM Limited. All rights reserved.

*

* @par

* ARM Limited (ARM) is supplying this software for use with Cortex-M

* processor based microcontrollers. This file can be freely distributed

* within development tools that are supporting such ARM based processors.

*

* @par

* THIS SOFTWARE IS PROVIDED "AS IS". NO WARRANTIES, WHETHER EXPRESS, IMPLIED

* OR STATUTORY, INCLUDING, BUT NOT LIMITED TO, IMPLIED WARRANTIES OF

* MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE APPLY TO THIS SOFTWARE.

* ARM SHALL NOT, IN ANY CIRCUMSTANCES, BE LIABLE FOR SPECIAL, INCIDENTAL, OR

* CONSEQUENTIAL DAMAGES, FOR ANY REASON WHATSOEVER.

Software Warranty Disclaimer

The author disclaim all warranties with regard to this software, including all implied warranties of merchantability and fitness. In no event shall the author be liable for any special, indirect or consequential damages or any damages whatsoever resulting from loss of use, data or profits, whether in an action of contract, negligence or other tortious action, arising out of or in connection with the use or performance of this software.

Hardware Disclaimer

This development board/kit is intended for use for FURTHER ENGINEERING, DEVELOPMENT, DEMONSTRATION, OR EVALUATION PURPOSES ONLY. It is not a finished product, and may not (yet) comply with some or any technical or legal requirements that are applicable to finished products, including, without limitation, directives regarding electromagnetic compatibility, recycling (WEEE), FCC, CE, or UL (except as may be otherwise noted on the board/kit). MattairTech LLC supplied this board/kit AS IS, without any warranties, with all faults, at the buyer's and further users' sole risk. The user assumes all responsibility and liability for proper and safe handling of the goods. Further, the user indemnifies MattairTech LLC from all claims arising from the handling or use of the goods. Due to the open construction of the product, it is the user's responsibility to take any and all appropriate precautions with regard to electrostatic discharge and any other technical or legal concerns.

The product described in this document is subject to continuous development and improvements. All particulars of the product and its use contained in this document are given by MattairTech LLC in good faith. However all warranties implied or expressed including but not limited to implied warranties of merchantability or fitness for particular purpose are excluded.

This document is intended only to assist the reader in the use of the product. MattairTech LLC shall not be liable for any loss or damage arising from the use of any information in this document or any error or omission in such information or any incorrect use of the product.

Export Compliance

If the item(s) are exported from the United States, then the purchaser will be screened using the following sources (other sources not listed here may also be used):

Department of Commerce – Bureau of Industry and Security
Denied Persons List - Unverified List - Entity List

Department of State – Bureau of International Security and Non-proliferation
Nonproliferation Sanctions

Department of State – Directorate of Defense Trade Controls
AECA Debarred List

Department of the Treasury – Office of Foreign Assets Control
Specially Designated Nationals List
Foreign Sanctions Evaders List
Sectoral Sanctions Identifications (SSI) List
Palestinian Legislative Council (PLC) List
The List of Foreign Financial Institutions Subject to Part 561 (the Part 561 List)
Non-SDN Iranian Sanctions Act List (NS-ISA)

If encryption hardware is present (L21 installed), then the following statements apply:

MattairTech LLC submits proper documentation to Bureau of Industry and Security (BIS) for classification and review. After a waiting period, the item can then be shipped worldwide.

This product contains encryption technology and is classified as ECCN 5A002A.1.
Exported under license exception ENC.

These commodities, technology, or software were exported from the United States in accordance with the export administration regulations. Diversion contrary to U.S. law is prohibited.

In some cases, MattairTech LLC will need to obtain further information from the purchaser. If so, we will send you an email with further instructions.

RoHS, FCC, CE, UL, WEEE, etc.

This development board/kit is intended for use for FURTHER ENGINEERING, DEVELOPMENT, DEMONSTRATION, OR EVALUATION PURPOSES ONLY. It is not a finished product, and may not (yet) comply with some or any technical or legal requirements that are applicable to finished products, including, without limitation, directives regarding electromagnetic compatibility, recycling (WEEE), FCC, CE, or UL (except as may be otherwise noted on the board/kit). This development board is not RoHS compliant.

Trademarks

Microchip®, Atmel®, and SAM-BA® are registered trademarks of Microchip Technology Inc.
ARM® and Cortex® are registered trademarks of ARM Ltd.
All other trademarks herein are property of their respective owners.

References

The microcontroller features / comparisons taken from respective Atmel datasheet:

D21: <http://ww1.microchip.com/downloads/en/DeviceDoc/40001882A.pdf>

L21: <http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-42385-SAM-L21-Datasheet.pdf>

C21: http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-42365-SAM-C21_Datasheet.pdf

Features list in Power Supply chapter from Microchip [datasheet](#)

Features list in Memory Device chapter from Microchip [datasheet](#)

Features list in Memory Device chapter from Adesto [datasheet](#)

Features list in Memory Device chapter from ON Semiconductor [datasheet](#)

Appendix A: Precautions

CAUTION

Do not change power configuration, or solder any jumper while unit is powered. Do not short Vin, Vbus, 3.3V, or ground to each other (ie: solder jumpers on bottom shorting on clipped lead).

CAUTION

Higher regulator input voltages mean larger voltage drops and thus higher thermal dissipation for a given amount of current. Be sure to limit current consumption to prevent excessive heat when using higher voltages and/or currents. The regulator will enter thermal shutdown if it gets too hot.

Note that the PTC fuse is located near the regulator, so high temperatures will lower the PTC trip and hold currents.

CAUTION


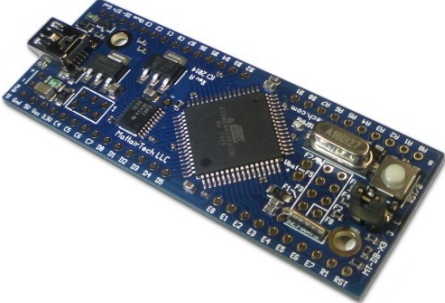
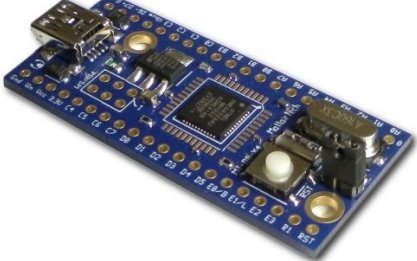
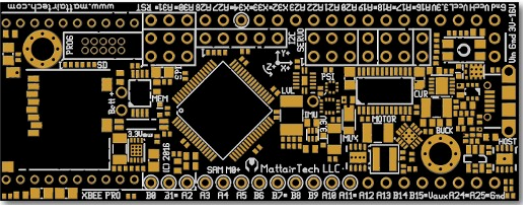
Normally, power is supplied from Vin or Vbus.

However, it is possible to disconnect the regulator and supply an externally regulated voltage on the 3.3V and/or Vcc pins. When doing this, care must be taken to limit inrush current on these pins due to the low ESR of the ceramic capacitors. Failure to do so may cause damaging inductive voltage spikes due to any wire inductance (ie: benchtop power supply leads). Inrush current is normally controlled by the PTC fuse, which has a small series resistance.

CAUTION

The MT-D21E contains static sensitive components.
Use the usual ESD procedures when handling.

Appendix B: Other MattairTech Products

	<p>MT-D11 USB ARM Cortex M0+ board</p> <ul style="list-style-type: none"> • ATSAM11D14AM (24-pin) • 16KB FLASH, 4KB SRAM • Onboard 3.3V, 250mA LDO regulator (2uA quiescent) • 16MHz and 32.768KHz crystals • USB connector (power by USB or external up to 16V) • Blue LED, 10-pin Cortex header, button, I2C pullups • USB CDC Bootloader (no programmer required) • Arduino 1.6.5+ support (core and bootloader)
	<p>MT-DB-X3 USB AVR XMEGA board</p> <ul style="list-style-type: none"> • XMEGA A3U, A3BU, C3, and D3 (64-pin) • 32KB - 384KB FLASH, 4KB – 32KB SRAM • 3.3V 250mA regulator (2uA quiescent current) • Optional 5V 500mA regulator (23uA quiescent current) • Optional auto-direction sensing level shifter • 16MHz and 32.768KHz crystals, optional coin cell holder • LED, boot jumper, PDI header, button, TWI pullups • USB DFU bootloader preinstalled (except D variant)
	<p>MT-DB-X4 USB AVR XMEGA board</p> <ul style="list-style-type: none"> • ATxmega128A4U USB XMEGA AVR • 128KB FLASH, 8KB SRAM, 2KB EEPROM • 3.3V LDO regulator (low quiescent current) • 16MHz and 32.768KHz crystals • LED, boot jumper, PDI header • Reset button, mounting holes • USB DFU bootloader preinstalled
	<p>D21J / L21J / C21J dev board coming soon!</p> <ul style="list-style-type: none"> • Choice of SAMD, SAML, or SAMC chips • XBEE / XBEE Pro socket (supports LTE modems) • MicroSD card slot / device and host mode USB • Motor controller / relay driver, up to 2.4A, 2.7V-16V • 2A buck converter with input up to 16V, power mux • Two 3.3V linear regulators (250mA and 1.5A w/enale) • SPI Serial memory (128KB SRAM with battery backup support on board, 64KB EEPROM, or 512KB FLASH) • 3-axis accelerometer / 3-axis gyroscope, pressure sensor / temperature sensor, current / voltage measure