SIKTEC

AVR Controller DIY – V.1.0.3

Introduction:

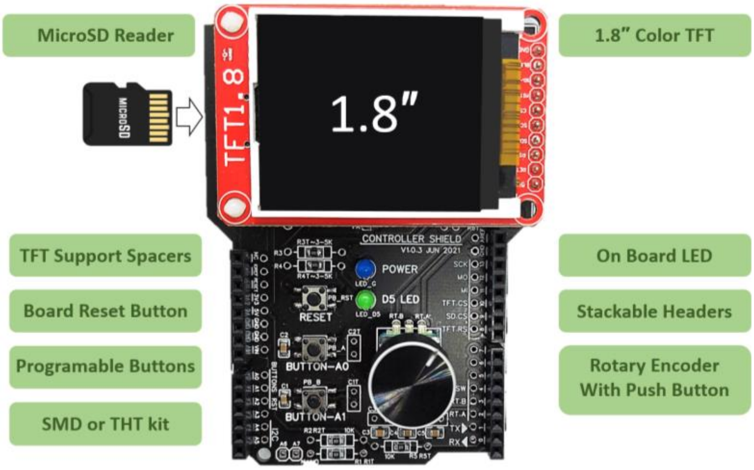
SIKTEC AVR Controller is a DIY Shield which extend most Arduino boards to easily integrate several common components which are widely used in any Arduino project who requires human interface. The shield features:

- Rotary Encoder with push button – Widely used to control menus and interfaces .
- 2 Programmable Tactile Push Buttons.
- 1 External Reset Button.
- 1.8' Colour TFT screen (ST7735) – Detachable with nylon board spacers.
- Micro SD card reader.
- All required headers to connect to the different Arduino Boards.

The shield supports SIKTEC Libraries and Example codes – All examples can be downloaded from GitHub: https://github.com/siktec-lab/SIKTEC-AVR-Controller.

Compatible and tested on:

- Arduino NANO V3.
- Arduino UNO – R3.
- Arduino LEONARDO.
- Arduino Mega.
- Arduino DUE.



Applications and Why?

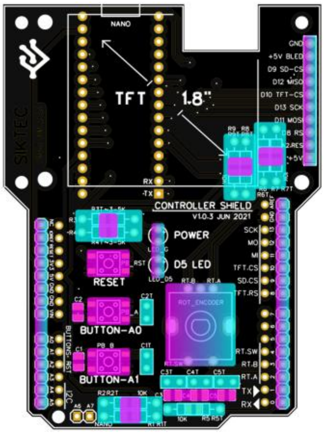
Our goal was to create a simple well designed shield which integrates widely used components in the best way. The shield enable makers to focus on what really matter – since most of the projects require those capabilities, makers "waste" there time on assembling and integrating common parts on breadboards and proto boards.

With this shield you can easily avoid all the hassle and focus on your idea. The shield is fully backed by our code base with simple and advanced examples – No need to google and search every where for the correct drivers and hack your way to make it work. All the code for this shield is well documented and can be easily extended and modified to fit your needs.

Feel free to reach out about any question, suggestion and feedback you have ☺.

Assembly Instruction:

- Follow the assembly order as described in the following table. This order is recommended to ease the assembly process.
- The shield support two types of parts – SMD & Through Hole. The marking on the board is duplicate for each part indicating which are SMD or Through Hole parts.
- The Coloured picture indicates where to place the Through Hole Parts (Cyan Colour) and where to place the SMD parts (Pink colour). The gradient colour mark parts which are mutual to both and must be assembled in both kits.

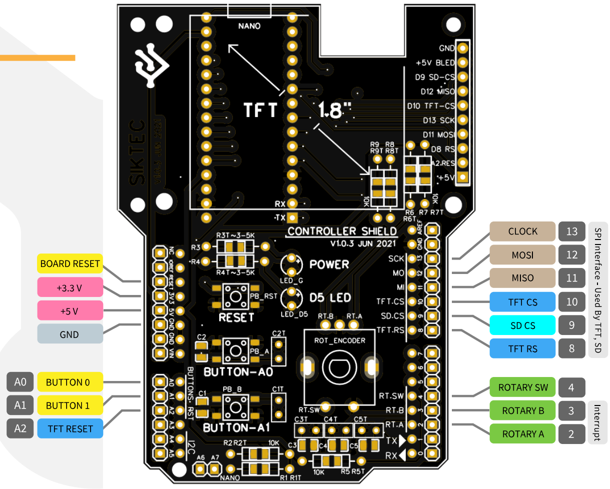


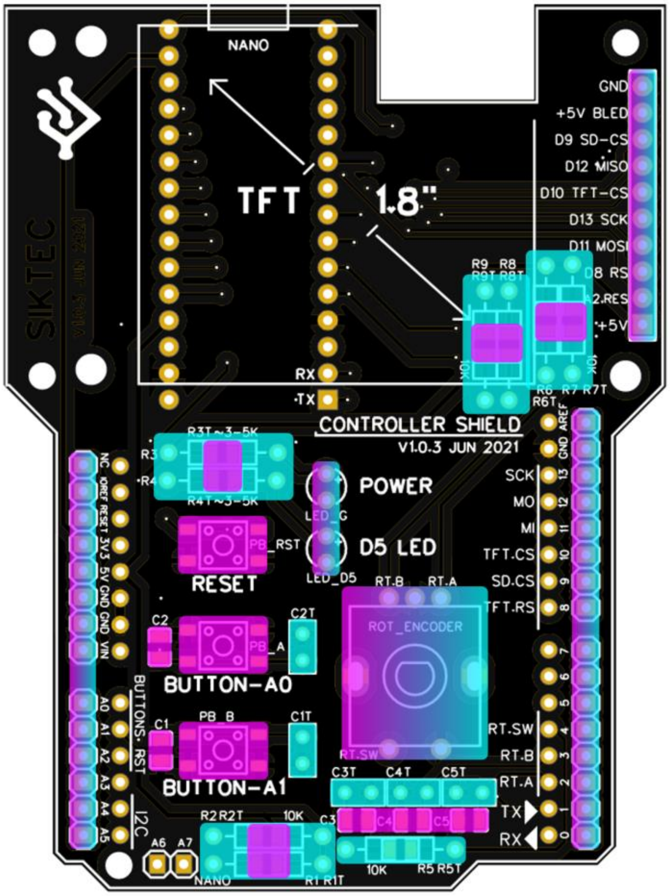
ORDER	**SMD** - MARKING	**THROUGH HOLE** - MARKING	VALUE
1	PB_RST, PB_A, PB_B	Pre-assembled	Tactile Buttons 4.5x4.5
2	C1, C2, C3, C4, C5	C1T, C2T, C3T, C4T, C5T	100nF
3	R1,R2,R5,R6,R7,R8,R9	R1T,R2T,R5T,R6T,R7T, R8T,R9T	10K Resistor
4 [1]	R3	R3T	47K Resistor
5 [1]	R4	R4T	20K Resistor
6 [2]	LED_G		Green LED
7 [2]	LED_B		Blue LED
8 [3]	Female Headers 10P, 3x8P, 6P		Use the form factor that fits your needs
9 [4]	ROT_ENCODER		12 pulse Encoder With Push Button Switch

1. Current limiting resistors for the LED's – Those are recommended values which are matching the supplied LEDs you can change those to make the LEDs dimmer – If you use your own LED's use other values who best fit your parts.
2. Feel free to use your own and change the colours – may also be omitted from the shield.
3. Female Headers should be assembled in regards of the shield propose.
   - NANO - Use the short female headers.
   - Arduino Boards (UNO, LEONARDO etc...) – You can choose between stacking headers (Long Legs) OR normal male pin connectors to keep it flush.
4. Rotary encoder is assembled last – This way you can easily align the connectors using a bread board. No matter which encoder you are using (Bourns, Cloned) the assembly is the same.

Controller Shield Pinout:

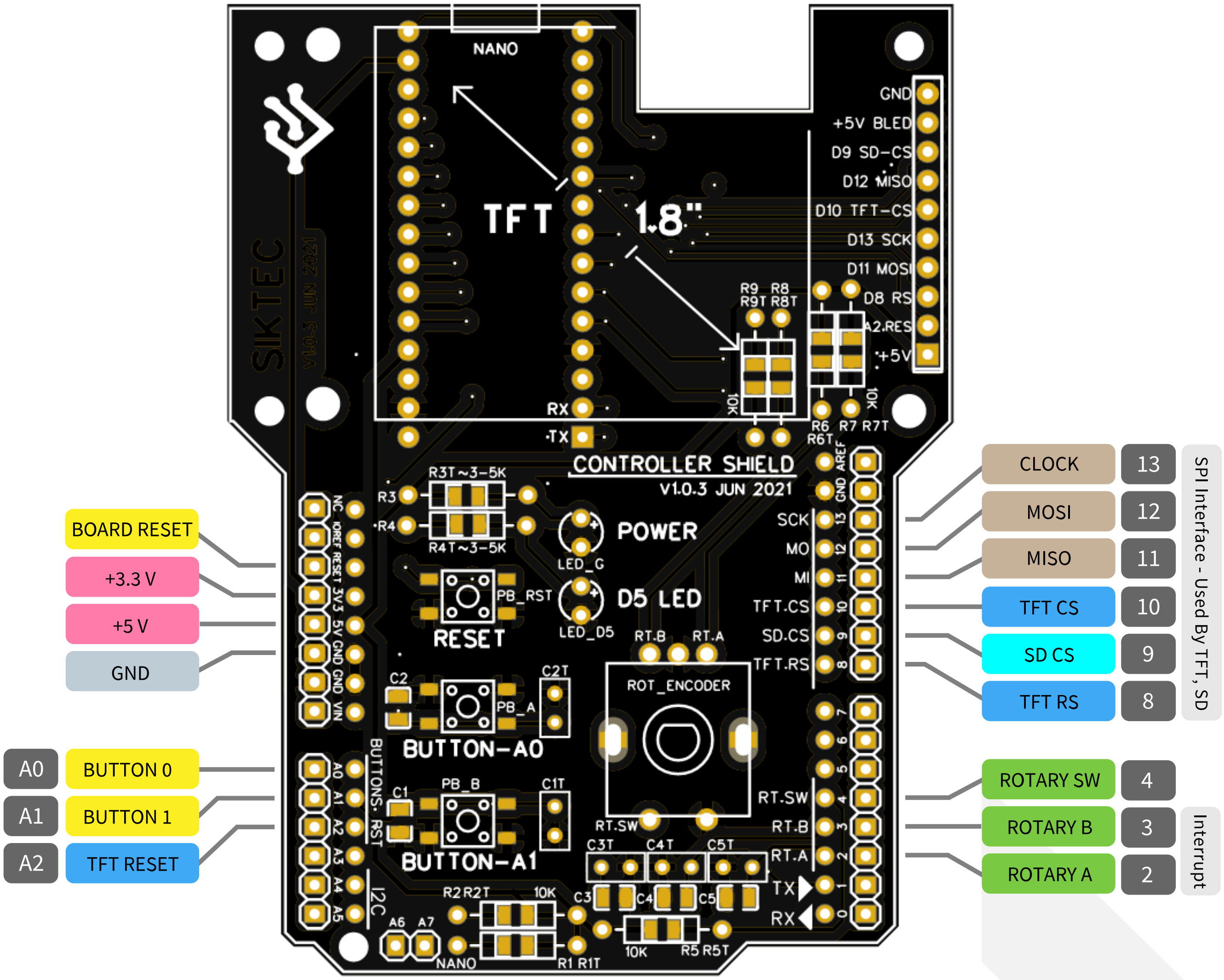
- The pinout of the shield is relative to Arduino UNO pins – No matter which board you use (Even NANO) The pinout is the same.
- Both the supplied 1.8 TFT, and the SD Card reader are attached to the Hardware SPI pins with the corresponding Chip Select pins as shown in the picture.
- Pin 2,3 (D2, D3) are used by the shield as the interrupt pins of the Encoder. The reason we design it Like this is to support Arduino UNO & NANO since those are the only raising Interrupts available.
- Rotary Push Button is connected to Pin 4 – You can see in the examples how to configure it as a change interrupt.
- This pinout is supported and set by SIKTEC_Rotary library which can be installed easily using the Arduino library manager or downloaded from GitHub and implemented in your projects: https://github.com/siktec-lab/SIK-Rotary-Encoder





ORDER	SMD - MARKING	THROUGH HOLE - MARKING	VALUE
1	PB_RST, PB_A, PB_B	Pre-assembled	Tactile Buttons 4.5x4.5
2	C1, C2, C3, C4, C5	C1T, C2T, C3T, C4T, C5T	100nF
3	R1,R2,R5,R6,R7,R8, R9	R1T,R2T,R5T,R6T,R7T, R8T,R9T	10K Resistor
4 [1]	R3	R3T	47K Resistor
5 [1]	R4	R4T	20K Resistor
6 [2]	LED_G		Green LED
7 [2]	LED_B		Blue LED
8 [3]	Female Headers 10P, 3x8P, 6P		Use the form factor that fits your needs
9 [4]	ROT_ENCODER		12 pulse Encoder With Push Button Switch

1. Current limiting resistors for the LED's – Those are recommended values which are matching the supplied LEDs you can change those to make the LEDs dimmer – If you use your own LED's use other values who best fit your parts.
2. Feel free to use your own and change the colours – may also be omitted from the shield.
3. Female Headers should be assembled in regards of the shield propose.
   - NANO - Use the short female headers.
   - Arduino Boards (UNO, LEONARDO etc...) – You can choose between stacking headers (Long Legs) OR normal male pin connectors to keep it flush.
4. Rotary encoder is assembled last – This way you can easily align the connectors using a bread board. No matter which encoder you are using (Bourns, Cloned) the assembly is the same.



Function	Pin
BOARD RESET	
+3.3 V	
+5 V	
GND	
BUTTON 0	A0
BUTTON 1	A1
TFT RESET	A2
CLOCK	13
MOSI	12
MISO	11
TFT CS	10
SD CS	9
TFT RS	8
ROTARY SW	4
ROTARY B	3
ROTARY A	2

SPI Interface - Used By TFT, SD

Interrupt