PCB Guide Part 3 - Creating the MCU schematic

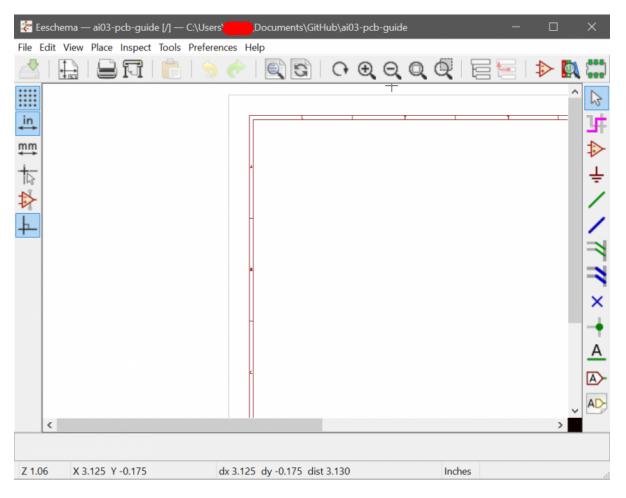
By this point, you should have installed the programs, created the repository, and added the local libraries.

Step 5. Beginning the schematic

The schematic is used to set up the electronics side of things.

For this tutorial, we will be using the typical Atmega32u4 microcontroller, and building a 2x2 macropad with MX and Alps support.

Open up the schematic editor to begin.



One modification used during this guide is seting the grid size to half the default size. This allows a bit finer grain control for wiring things.

From Preferences -> Preferences -> Display options, set the grid size to 10.

Preferences				×
Preferences Common Hotkeys Eeschema Display Options Colors Field Name Templates	Grid Options (not supported in Legacy Toolset) Grid Style Dots Lines Small crosses Grid thickness: 1	Dimensions Bus thickness: Line thickness: Junction size: Annotations Symbol unit no Show hidde Show page	40.000 otation: A	mils mils
			ОК	Cancel

Also, if you want the super HD 8K experience while making your schematic, you can swap the graphics settings in the Common page.

Preferences		×
Common Hotkeys Eeschema Display Options Colors Field Name Templates	Auto save: 10 File history size: 9 Graphics (Accelerated): Supersampling (4x) Graphics (Fallback): No Antialiasing Subpixel Antialiasing (High Quality) Helper Applications Subpixel Antialiasing (Ultra Quality) Text editor: C:\Progra Supersampling (2x) • System default PDF viewer • • Other: • User Interface 275 Icon scale: 10 • Show icons in menus Pan and Zoom • Center and warp cursor on zoom • Use touchpad to pan • Pan while moving object	
	ОК	Cancel

From Ruiqi Mao's PCB guide, here is a nice commands reference:

m: pick the component up and move it
g: drag the component up and move it while keeping wires attached to it
c: copy the component
e: edit the component
r: rotate the component
y: mirror the component
del: delete the component
esc: abort!

Also, these shortcuts are useful to know:

w - Begin drawing a wire connection

k - Cut a wire and stop drawing it without clicking on an endpoint

Ctrl + h - Place a global net

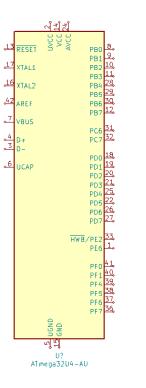
The microcontroller, or MCU for short, is the brain of the PCB. It cannot function on its own, so we will build the circuit for it.

First, press A to open the "add symbol" menu, and search for atmega32u4.

Select the atmega32u4-au component.

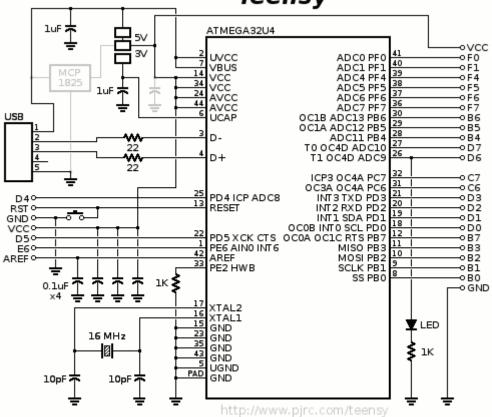
Choose Symbol (12185 items loaded)			×		
atmega32u4		। ্ৰিবি			
Symbol History ATmega32U4-AU MCU_Microchip_ATmega ATmega32U4-AU ATmega32U4-MU ATmega32U4RC-AU ATmega32U4RC-MU	Desc Recently u 16MHz, 32 Microchip 16MHz, 32 16MHz, 32 16MHz, 32 16MHz, 32	U - 시크(로) 13 RESE 3 5 5 5 5 17 YIALL 15 YIAL2 23 AREF 2 YEUS 4 D. 24 D. 25 UCAP	F80 8. F81 10.1 F83 10.1 P83 11.1 P85 23.2 P85 12.1 P65 12.1 P70 16.0 P10 10.1 P10 20.1 P10 20.1 P10 20.1 P10 21.1 P10 21.1		
Alias of ATmega16U4-AU (16MHz, 16kB Flash, 1.25kB SRAM, 512B EEPROM, USB 2.0, TQFP-44)					
16MHz, 32kB Flash, 2.5kB SRAM, 1kB EEPR Key words: AVR 8bit Microcontroller MegaA Reference U? Value ATmega16U4-AU <		+-+	>		
		ОК	Cancel		

Place it down in the schematic.



To set it up, we will simply follow two resources:

- The technical data provided by the official datasheet. http://ww1.microchip.com/downloads/en/devicedoc/atmel-7766-8-bit-avr-atmega16u4-32u4_datasheet.pdf
- An existing "tried and tested" schematic used on the PJRC Teensy.

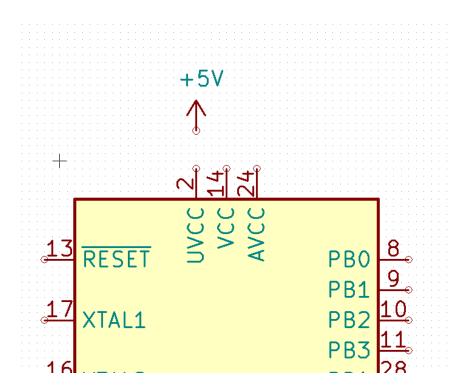


Teensy

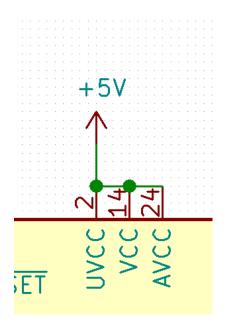
Usually, both a technical datasheet and an existing implementation will exist for most components. Search for both, and use both as reference.

To begin, let's tie UVCC, VCC, and AVCC to +5V, as specified by the datasheet. Hit P to open the power symbols menu, and select the +5V symbol.

Place it above the VCC pins. If you misplace, you can move it by hovering over the component and pressing M.

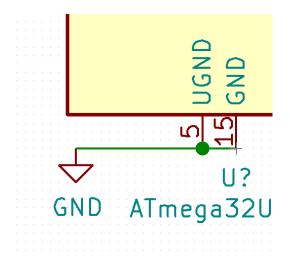


Now, wire it together using the wire tool (hotkey W).



This binds the physical UVCC/VCC/AVCC pins on the component to the +5V net. Net refers to a set of pads that should all be connected together.

Similarly, bind the GND and UGND pins to Ground by selecting the GND symbol in the power menu and wiring them together.



Protip: It is good practice to have positive power source symbols pointing up, and ground power symbols pointing down.

Now we will place some components.

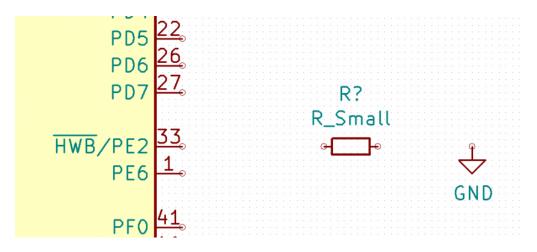
The HWB pin requires a pull-down resistor, or a resistor connected to ground.

Open the components menu with A, and search for r_small, or the small resistor component (The large one is massive and will burn space on the schematic):

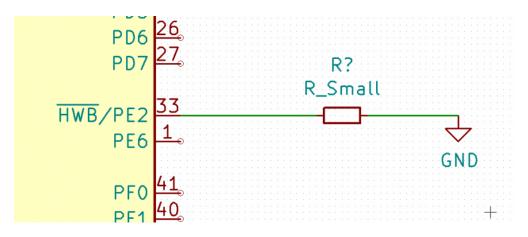
Choose Symbol (12185 items loaded)		×
r_small		
Symbol Device	Desc Generic de	
R_Small	Resistor, s	
D_Zener_Small D Zener Small ALT	Zener Dioc Zener Dioc	Ĥ_
Resonator_Small	Three pin (R_Small
📄 Graphic	Graphical s	
SYM_LASER_Small	small LASE	•

Now, place this parallel to the pin labeled HWB/PE2. You can rotate the component by pressing R.

Also, place a ground symbol.



Now, wire it together.

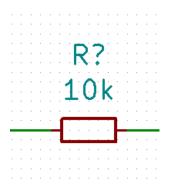


Congratulations, now the PE2 pin is connected to ground through the resistor.

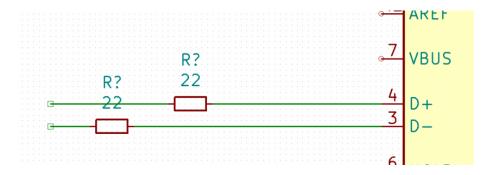
Let's change the value of the resistor. Press the E key while hovering over the resistor to edit. A 10k ohm resistor is fit for this task. Select the value field, and type 10k.

Symbol Properties				- 🗆	\times
Unit:	Fields:				
A v	Name	Value	Horizontal Position:	Vertical Positio	n:
Orientation (degrees):	Reference		O Align left	O Align top	
0	Value	10k	Align center	Align cente	r
0 +90	For print		O Align right	O Align botto	m
0 11	Ditasheet	~			
○ +180	•				
• -90			Visibility:	Font Style:	
Aspect:			Show	Normal	
 Default 			Rotate	Oltalic	
O Mirror around X axis				OBold	
O Mirror around Y axis				O Bold and its	alic
Convert shape			Field Name:		
Library Symbol:			Value		
Device:R_Small		×	Field Value:		
Validate Change			10k		
Symbol ID:					
5C33F45C			Font size: 0.050		in
			Position X: -0.105		in
Edit Spice Model			Position Y: 0.000		in
Reset Field Properties	<	>			
Update Field Values		- +			
			0	K Can	cel

The component's value should now be updated.



The USB pins also need resistors. Repeat the previously described commands to place two 22 ohm resistors, and wire them.



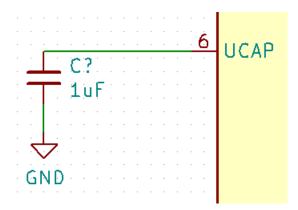
In this case, I drew the wire out on the other side as well, and used the K hotkey to end the wire at those points.

Now, let's place needed capacitors.

Through the add components menu, search for c_small (Again, regular capacitor symbol is massive).

Choose Symbol (12186 items loaded)		×
c_small		
Symbol Device C_Small Jumper_NC_Small	Desc Generic de Unpolarize Jumper, no	C C_Small
<	>	
C_Small Unpolarized capacitor Key words: capacitor cap		^

First, place a capacitor with value 1uF beside the UCAP pin, and connect the other end to ground.



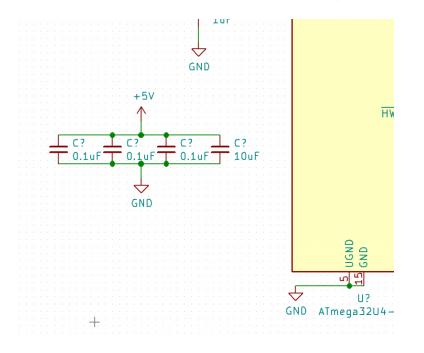
Next, we want to set up our decoupling capacitors. These are important capacitors that filter out electrical noise coming from the microcontroller.

The official documentation states to have a 0.1uF by each VCC pin (Which would be 4 on a 32u4), and a 10uF overall.

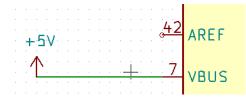
However, the Teensy schematic uses only three 0.1uF, and a 1uF overall.

I personally combine them - three 0.1uF and one 10uF. This has been confirmed functional by my testing. Place them in an empty space close to the microcontroller, and bind one end to +5V, one end to Gnd.

Protip: Build one 0.1uF capacitor, and clone it using C to save time.



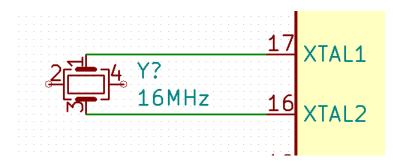
Since this PCB will get its power from USB no matter, bind VBUS to +5V.



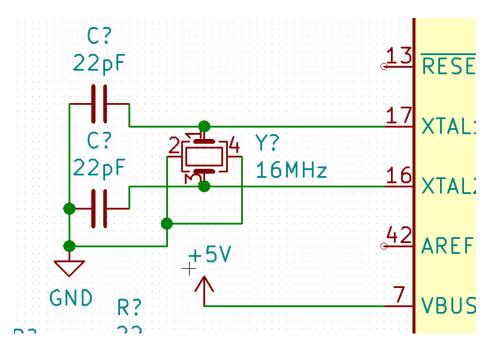
The crystal controls how fast the controller functions. Search for the Crystal_GND24_Small component...

Choose Symbol (12187 items loaded)		
crystal_gnd24_small		
Symbol	Desc	
Device	Generic de	
Crystal_GND24_Small	Four pin a	
		۹ ^۲ CrystaLGN ماریک

...and wire the non-gnd pins (1 and 3) to XTAL1 and XTAL2.

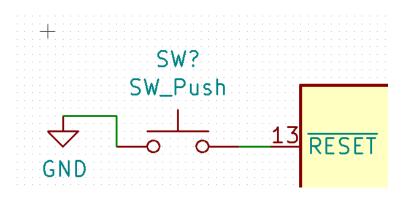


The crystal also needs load capacitors of its own. Place two 22pF capacitors between each crystal pin and Gnd. Also, wire the Gnd pins on the crystal to Gnd.



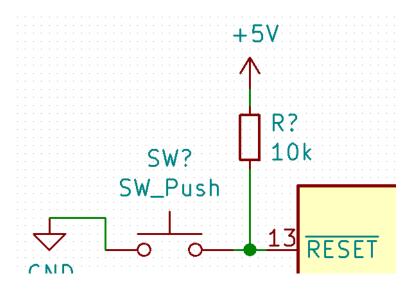
The reset button is pressed to make the MCU enter bootloader mode. This is used to write the firmware to the PCB.

Place a sw_push component between reset and ground.



Also, an external pull-up resistor can prevent the reset pin from being read faultily, and accidentally resetting during use.

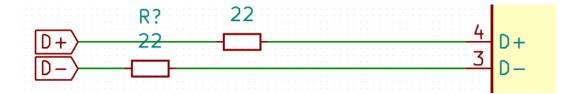
For this, a 10k ohm resistor can be placed between reset and +5V.



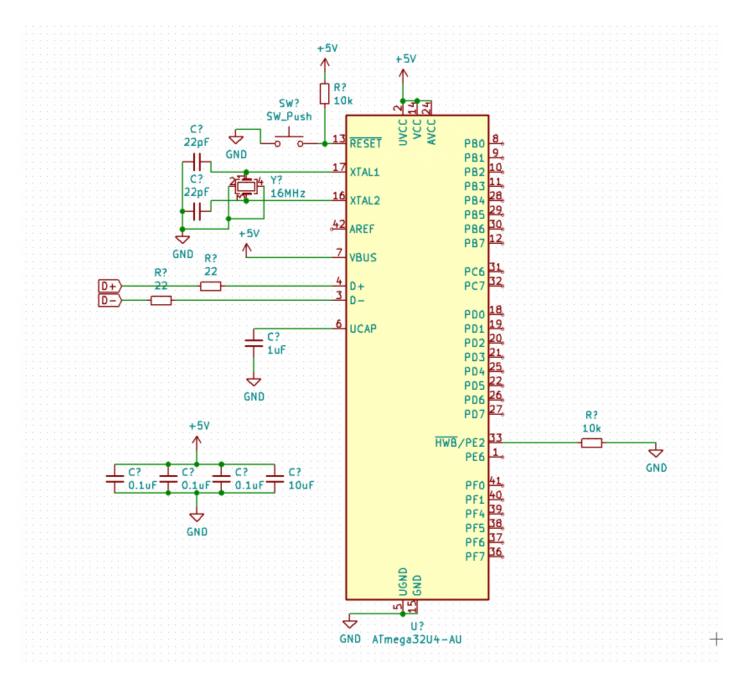
The AREF pin is ignored since we won't be doing analog readings.

Now let's place global nets on the USB lines - These connect wires together by name, removing the need for ugly wires that cut across the entire schematic.

Hit Ctrl+H, type the names, and place them on the other end of the USB positions.



Local nets also work, but I prefer global nets since they are much more visible.



A view of the completed MCU zone.

Now, it's time to complete the rest.

Revision #17 Created 8 January 2019 00:24:22 by ai03 Updated 29 June 2020 01:48:46 by ai03