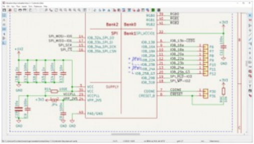
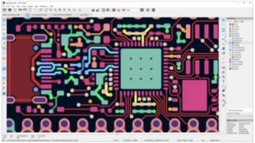
Module 15 - In the Wild

KiCad: Breadboard are good, but PCBs are better

Schematic Capture

KiCad's Schematic Editor supports everything from the most basic schematic to a complex hierarchical design with hundreds of sheets. Create your own custom symbols or use some of the thousands found in the official KiCad library. Verify your design with integrated SPICE simulator and electrical rules checker.



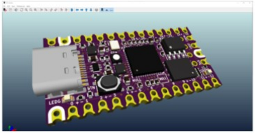


PCB Layout

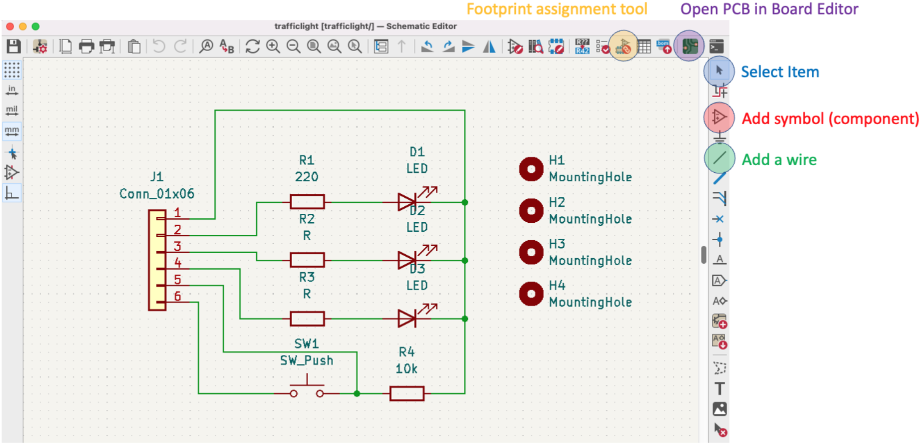
KiCad's PCB Editor is approachable enough to make your first PCB design easy, and powerful enough for complex modern designs. A powerful interactive router and improved visualization and selection tools make layout tasks easier than ever.

3D Viewer

KiCad's 3D Viewer allows easy inspection of your PCB to check mechanical fit and to preview your finished product. A built-in raytracer with customizable lighting can create realistic images to show off your work.



Start with schematics

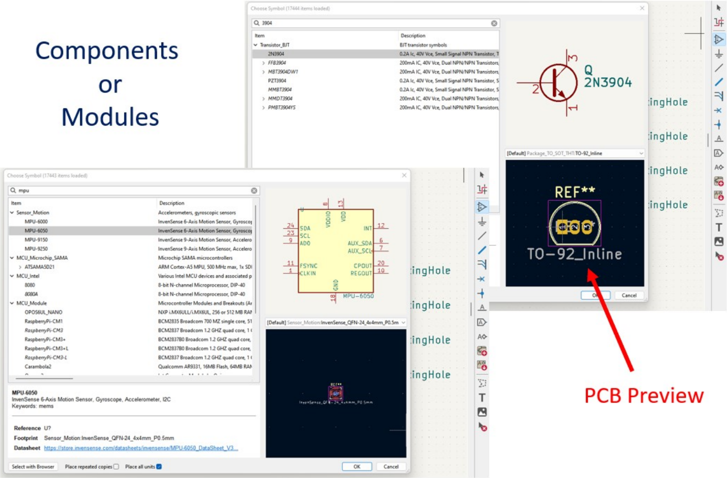


Rich library of components

Components

or

Modules



Choose Symbol (17444 items loaded)

🔍 3904

Item	Description
⌄ Transistor_BJT	BJT transistor symbols
2N3904	0.2A Ic, 40V Vce, Small Signal NPN Transistor, T
› FFB3904	200mA IC, 40V Vce, Dual NPN/NPN Transistors,
› MBT3904DW1	200mA IC, 40V Vce, Dual NPN/NPN Transistors,
PZT3904	0.2A Ic, 40V Vce, Small Signal NPN Transistor, S
MMBT3904	0.2A Ic, 40V Vce, Small Signal NPN Transistor, S
› MMDT3904	200mA IC, 40V Vce, Dual NPN/NPN Transistors,
› PMBT3904YS	200mA IC, 40V Vce, Dual NPN/NPN Transistors,

[Default] Package_TO_SOT_THT:TO-92_Inline

Choose Symbol (17443 items loaded)

🔍 mpu

Item	Description
⌄ Sensor_Motion	Accelerometers, gyroscopic sensors
MPU-6000	InvenSense 6-Axis Motion Sensor, Gyroscop
MPU-6050	InvenSense 6-Axis Motion Sensor, Gyroscop
MPU-9150	InvenSense 9-Axis Motion Sensor, Accelero
MPU-9250	InvenSense 9-Axis Motion Sensor, Accelero
⌄ MCU_Microchip_SAMA	Microchip SAMA microcontrollers
› ATSAMA5D21	ARM Cortex-A5 MPU, 500 MHz max, 1x SDI
⌄ MCU_Intel	Various Intel MCU devices and associated p
8080	8-bit N-channel Microprocessor, DIP-40
8080A	8-bit N-channel Microprocessor, DIP-40
⌄ MCU_Module	Microcontroller Modules and Breakouts (A
OPOS6UL_NANO	NXP i.MX6ULL/i.MX6UL, 256 or 512 MB RAM
RaspberryPi-CM1	BCM2835 Broadcom 700 MZ single core, 51
RaspberryPi-CM3	BCM2837 Broadcom 1.2 GHZ quad core, 1
RaspberryPi-CM3+	BCM2837B0 Broadcom 1.2 GHZ quad core,
RaspberryPi-CM3+L	BCM2837B0 Broadcom 1.2 GHZ quad core,
RaspberryPi-CM3-L	BCM2837 Broadcom 1.2 GHZ quad core, 1
Carambola2	Qualcomm AR9331, 16MB Flash, 64MB RAM

MPU-6050
InvenSense 6-Axis Motion Sensor, Gyroscope, Accelerometer, I2C
Keywords: mems

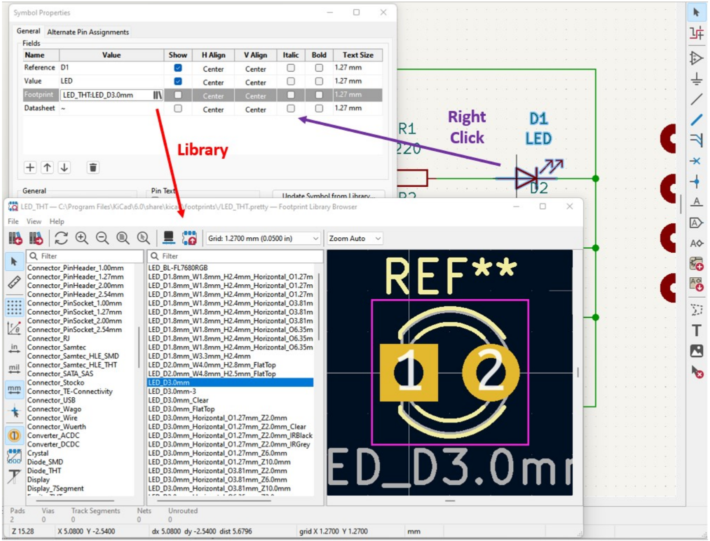
Reference U?
Footprint Sensor_Motion:InvenSense_QFN-24_4x4mm_P0.5mm
Datasheet https://store.invensense.com/datasheets/invensense/MPU-6050_DataSheet_V3...

[Default] Sensor_Motion:InvenSense_QFN-24_4x4mm_P0.5m

Select with Browser   Place repeated copies ☐   Place all units ☑

PCB Preview

Select a footprint for each component



Symbol Properties

General | Alternate Pin Assignments

Fields

Name	Value	Show	H Align	V Align	Italic	Bold	Text Size
Reference	D1	☑	Center	Center	□	□	1.27 mm
Value	LED	☑	Center	Center	□	□	1.27 mm
Footprint	LED_THT:LED_D3.0mm	□	Center	Center	□	□	1.27 mm
Datasheet	~	□	Center	Center	□	□	1.27 mm

LED_THT — C:\Program Files\KiCad\6.0\share\kicad\footprints\LED_THT.pretty — Footprint Library Browser

Grid: 1.2700 mm (0.0500 in)

Zoom Auto

Connector_PinHeader_1.00mm
Connector_PinHeader_1.27mm
Connector_PinHeader_2.00mm
Connector_PinHeader_2.54mm
Connector_PinSocket_1.00mm
Connector_PinSocket_1.27mm
Connector_PinSocket_2.00mm
Connector_PinSocket_2.54mm
Connector_RJ
Connector_Samtec
Connector_Samtec_HLE_SMD
Connector_Samtec_HLE_THT
Connector_SATA_SAS
Connector_Stocko
Connector_TE-Connectivity
Connector_USB
Connector_Wago
Connector_Wire
Connector_Wuerth
Converter_ACDC
Converter_DCDC
Crystal
Diode_SMD
Diode_THT
Display
Display_7Segment

LED_BL-FL7680RGB
LED_D1.8mm_W1.8mm_H2.4mm_Horizontal_O1.27m
LED_D1.8mm_W1.8mm_H2.4mm_Horizontal_O1.27m
LED_D1.8mm_W1.8mm_H2.4mm_Horizontal_O1.27m
LED_D1.8mm_W1.8mm_H2.4mm_Horizontal_O3.81m
LED_D1.8mm_W1.8mm_H2.4mm_Horizontal_O3.81m
LED_D1.8mm_W1.8mm_H2.4mm_Horizontal_O3.81m
LED_D1.8mm_W1.8mm_H2.4mm_Horizontal_O6.35m
LED_D1.8mm_W1.8mm_H2.4mm_Horizontal_O6.35m
LED_D1.8mm_W1.8mm_H2.4mm_Horizontal_O6.35m
LED_D1.8mm_W3.3mm_H2.4mm
LED_D2.0mm_W4.0mm_H2.8mm_FlatTop
LED_D2.0mm_W4.8mm_H2.5mm_FlatTop
LED_D3.0mm
LED_D3.0mm-3
LED_D3.0mm_Clear
LED_D3.0mm_FlatTop
LED_D3.0mm_Horizontal_O1.27mm_Z2.0mm
LED_D3.0mm_Horizontal_O1.27mm_Z2.0mm_Clear
LED_D3.0mm_Horizontal_O1.27mm_Z2.0mm_IRBlack
LED_D3.0mm_Horizontal_O1.27mm_Z2.0mm_IRGrey
LED_D3.0mm_Horizontal_O1.27mm_Z6.0mm
LED_D3.0mm_Horizontal_O1.27mm_Z10.0mm
LED_D3.0mm_Horizontal_O3.81mm_Z2.0mm
LED_D3.0mm_Horizontal_O3.81mm_Z6.0mm
LED_D3.0mm_Horizontal_O3.81mm_Z10.0mm

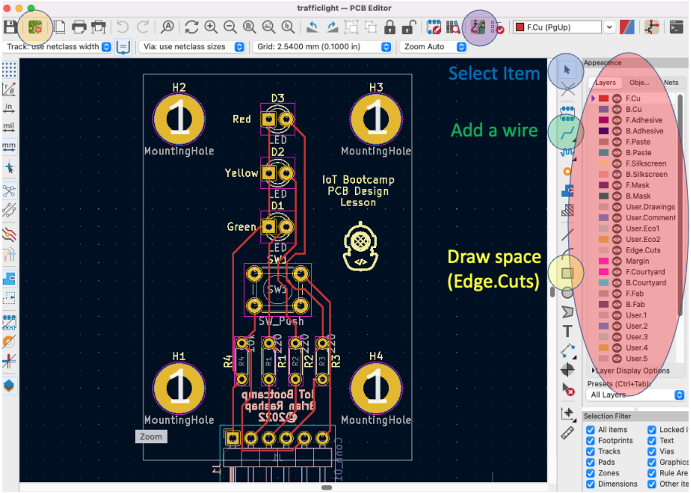
Pads	Vias	Track Segments	Nets	Unrouted
2	0	0	0	0

Z 15.28 X 5.0800 Y -2.5400 dx 5.0800 dy -2.5400 dist 5.6796 grid X 1.2700 Y 1.2700 mm

Generate PCB, Route Wires

Edit board setup including layers, design rules, and various defaults

Update PCB with changes make to schematic



Track: use netclass width

Via: use netclass sizes

Grid: 2.5400 mm (0.1000 in)

Zoom Auto

F.Cu (PgUp)

Select Item

Add a wire

Draw space (Edge.Cuts)

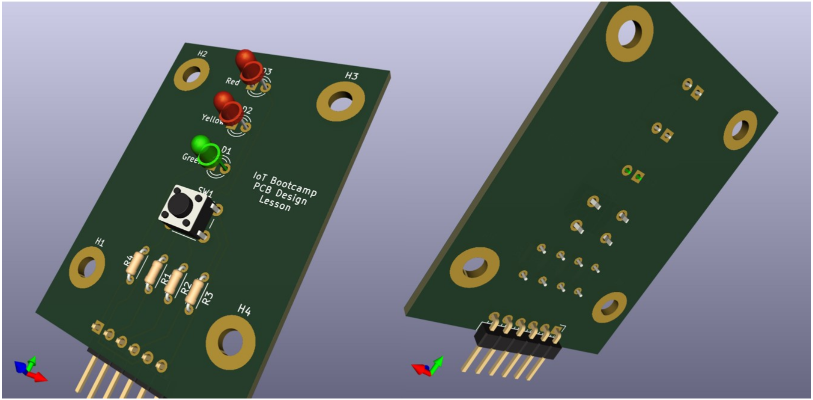
Selection Filter

☑ All items ☑ Locked i
☑ Footprints ☑ Text
☑ Tracks ☑ Vias
☑ Pads ☑ Graphics
☑ Zones ☑ Rule Are
☑ Dimensions ☑ Other ite

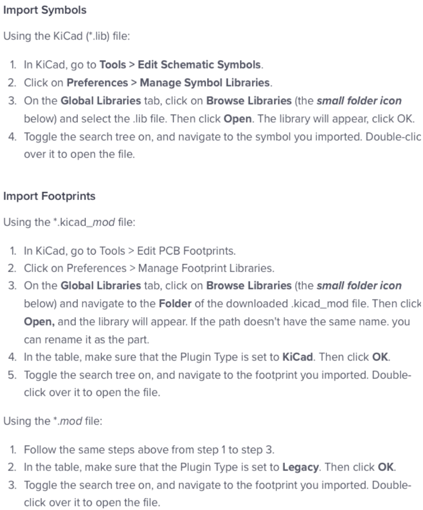
Presets (Ctrl+Tab):
All Layers

Layers (including hide/unhide)

3D View


H2
Red
D3
H3
Yellow
D2
Green
D1
IoT Bootcamp
PCB Design
Lesson
SW1
H1
R4
R1
R2
R3
H4

Import into KiCad



Import Symbols

Using the KiCad (*.lib) file:

1. In KiCad, go to **Tools > Edit Schematic Symbols**.
2. Click on **Preferences > Manage Symbol Libraries**.
3. On the **Global Libraries** tab, click on **Browse Libraries** (the ***small folder icon*** below) and select the .lib file. Then click **Open**. The library will appear, click OK.
4. Toggle the search tree on, and navigate to the symbol you imported. Double-clic over it to open the file.

Import Footprints

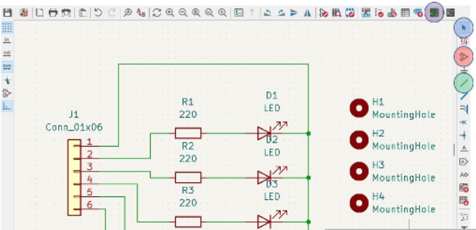
Using the *.kicad_*mod* file:

1. In KiCad, go to Tools > Edit PCB Footprints.
2. Click on Preferences > Manage Footprint Libraries.
3. On the **Global Libraries** tab, click on **Browse Libraries** (the ***small folder icon*** below) and navigate to the **Folder** of the downloaded .kicad_mod file. Then click **Open,** and the library will appear. If the path doesn't have the same name. you can rename it as the part.
4. In the table, make sure that the Plugin Type is set to **KiCad**. Then click **OK**.
5. Toggle the search tree on, and navigate to the footprint you imported. Double-click over it to open the file.

Using the *.*mod* file:

1. Follow the same steps above from step 1 to step 3.
2. In the table, make sure that the Plugin Type is set to **Legacy**. Then click **OK**.
3. Toggle the search tree on, and navigate to the footprint you imported. Double-click over it to open the file.

Assignment: L15_TrafficPCB

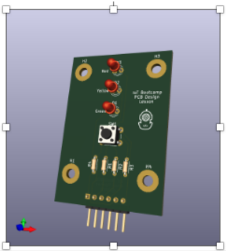


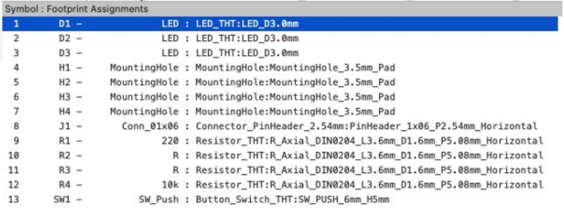
Open PCB in Board Editor

Select item

Add symbol (component)

Add a wire

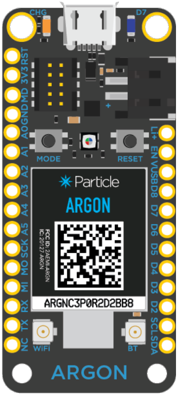




Symbol : Footprint Assignments

1	D1 –	LED :	LED_THT:LED_D3.0mm
2	D2 –	LED :	LED_THT:LED_D3.0mm
3	D3 –	LED :	LED_THT:LED_D3.0mm
4	H1 –	MountingHole :	MountingHole:MountingHole_3.5mm_Pad
5	H2 –	MountingHole :	MountingHole:MountingHole_3.5mm_Pad
6	H3 –	MountingHole :	MountingHole:MountingHole_3.5mm_Pad
7	H4 –	MountingHole :	MountingHole:MountingHole_3.5mm_Pad
8	J1 –	Conn_01x06 :	Connector_PinHeader_2.54mm:PinHeader_1x06_P2.54mm_Horizontal
9	R1 –	220 :	Resistor_THT:R_Axial_DIN0204_L3.6mm_D1.6mm_P5.08mm_Horizontal
10	R2 –	R :	Resistor_THT:R_Axial_DIN0204_L3.6mm_D1.6mm_P5.08mm_Horizontal
11	R3 –	R :	Resistor_THT:R_Axial_DIN0204_L3.6mm_D1.6mm_P5.08mm_Horizontal
12	R4 –	10k :	Resistor_THT:R_Axial_DIN0204_L3.6mm_D1.6mm_P5.08mm_Horizontal
13	SW1 –	SW_Push :	Button_Switch_THT:SW_PUSH_6mm_H5mm

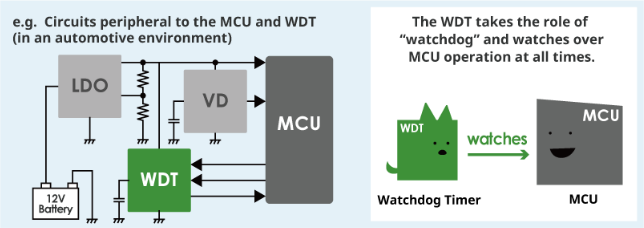
EN and RST pins



The Argon has two pins that are extremely useful in real world situations:

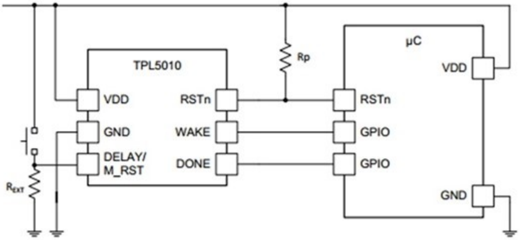
1. EN pin
   - The EN pin is not a power pin, per se, but it controls the 3V3 power.
   - Device enable pin is internally pulled-up. To disable the device (force the device into a deep power-down state), connect this pin to GND.
   - This pin is essentially an on/off pin.
2. RST pin
   - Active-low system reset input. This pin is internally pulled-up.

Watchdog Timer


e.g. Circuits peripheral to the MCU and WDT
(in an automotive environment)
LDO
VD
MCU
WDT
12V Battery
The WDT takes the role of "watchdog" and watches over MCU operation at all times.
WDT
watches
MCU
Watchdog Timer
MCU

The watchdog timer communicates with the MCU at a set interval. If the MCU does not output a signal, outputs too many signals or outputs signals that differ from a predetermined pattern, the timer determines that the MCU is malfunctioning and sends a reset signal to the MCU.

Hardware Watchdog Timers - TPL5010



The timeout frequency is set by resistor R_{EXT} using a formula from the data sheet.

Timeout Interval	Calculated Resistance
1 minute	22 Ω
5 minutes	43 Ω
30 minutes	92 Ω
1 hour	125 Ω
2 hours	170 Ω

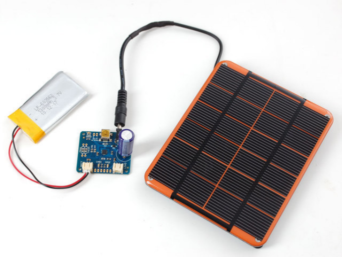
Application Watchdog

The Argon also has a watchdog timer that can be implemented in code. It is not as robust as the hardware timer, but better than nothing.

```
// Prototype
// ApplicationWatchdog(unsigned timeout_ms, std::function<void(void)> fn, unsigned stack_size=DEFAULT_STACK_SIZE);

// Global variable to hold the watchdog object pointer
ApplicationWatchdog *wd;

void watchdogHandler() {
  // Do as little as possible in this function, preferably just a reset
  System.reset(RESET_NO_WAIT);
}

void setup() {
  // Start watchdog. Reset the system after 60 seconds if the application is unresponsive
  // The stack_size default is 512, but this is too small. Use at least 1536.
  wd = new ApplicationWatchdog(60000, watchdogHandler, 1536);
}

void loop() {
  while (some_long_process_within_loop) {
    ApplicationWatchdog::checkin(); // resets the AWDT count
  }
}
// AWDT count reset automatically after loop() ends
```

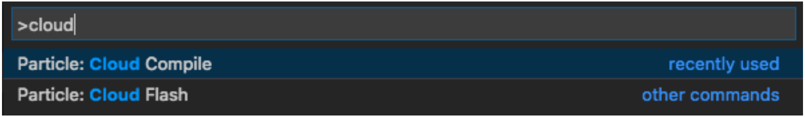
Solar Charging

Should be easy, but isn't.



Cloud Flash

During the deployment phase, it isn't convenient to have to hook up a USB cable to push updates. The Particle ecosystem allows for sending code over-the-air (OTA).



```
>cloud
Particle: Cloud Compile                                        recently used
Particle: Cloud Flash                                         other commands
```

- Cloud Compile - compile your program and download the binary
- Cloud Flash - compile and flash it to the selected device OTA

The OTA operations require:

- The device is connected to the Particle Cloud (breathing cyan)
- The computer is into the same account that claimed the device.
- The DeviceID is set to the device name.