**DIY pocket thermal imager**

https://hackaday.io/project/189728-diy-pocket-thermal-imager

**Components**

* 1 × Melexis MLX90640 sensor (or EVB90640-41 from DigiKey) – *Thermal sensor*
* 1 × LILYGO TTGO T4 v1.3 board – *Microcontroller and display unit*
* 1 × 3.7V 1800mAh 103449 LiPo Battery – *Powers the device (get as thin as possible)*
* 1 × JST 1.25mm 2 pin connector plug – *Battery connector*
* 1 × 280-point perfboard (thin) – *Board for components*
* 1 × 2x10 2.54mm pin connector – *To connect the TTGO T4 to the perfboard*
* 2 × 4.7 kΩ resistors – *Pull-up resistors for both SDA and SCL lines*
* 1 × 4.7 kΩ resistor – *For measuring battery voltage (IO35 to GND)*
* 1 × 5.1 kΩ resistor – *For measuring battery voltage (IO35 to VBAT)*
* 1 × 4.7 µF capacitor – *Low pass noise filter for VDD and GND (optional)*
* 1 × 1.0 µF capacitor – *Low pass noise filter for VDD and GND (optional)*
* 1 × 0.1 µF capacitor – *Low pass noise filter for VDD and GND (optional)*
* 2 × 2mm shrink wrap – *For sealing soldered wires to battery*
* 24 AWG wire – *To run SDA and SCL to pins*
* 1 × 0.3 mm solder
* 1 × 0.6 mm solder

**Tools**

* Fine point soldering iron
* Multimeter to check voltage
* Micro-USB cable (for programming & charging – must be able to pass data to computer)
* Espressif idf 4.4.3 (bash based software for flashing device)

**MLX90640 design:**

|  |  |  |
| --- | --- | --- |
| Pin # | Name | Description |
| 1 | SDA | I2C serial data (input/output) |
| 2 | VDD | Positive supply |
| 3 | GND | Negative supply (Ground) |
| 4 | SCL | I2C serial clock (input only) |

A diagram of a circular object

AI-generated content may be incorrect.

**LILYGO T4 v1.3 pin map:**

A diagram of a circuit board

AI-generated content may be incorrect.

\*This uses the ILI9341 driver, double check code

SPI connections:

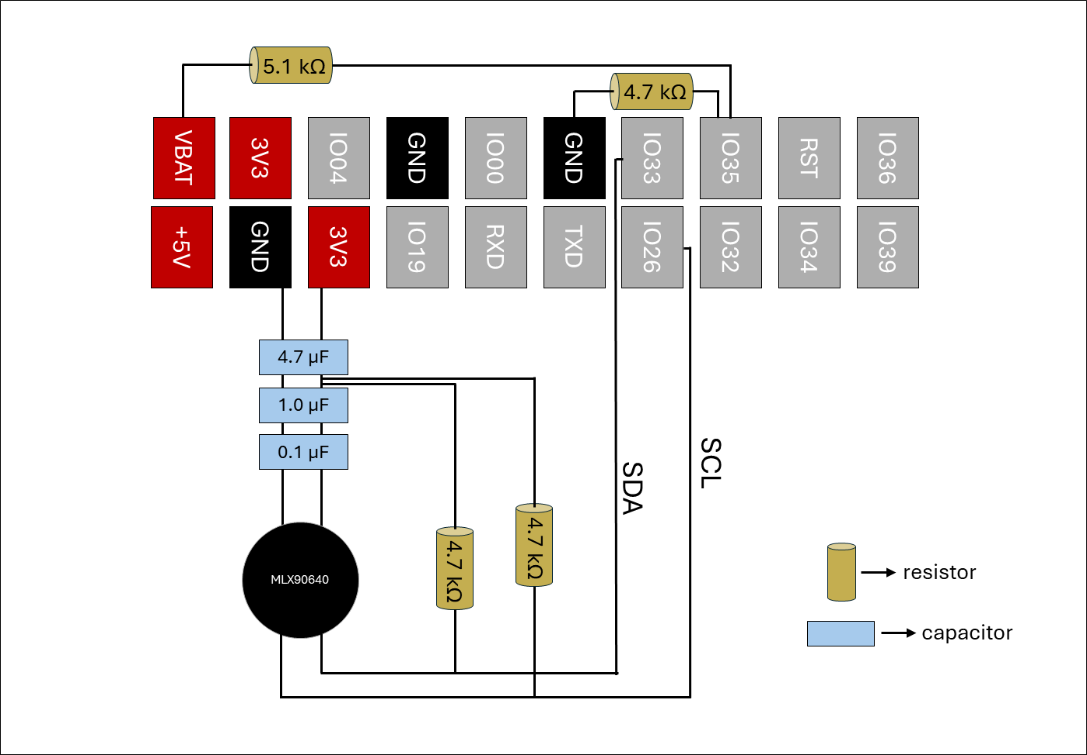
|  |  |  |
| --- | --- | --- |
| SD Card Pin | ESP32 GPIO | Notes |
| MISO | GPIO2 | Data from SD card to ESP32 |
| MOSI | GPIO15 | Data from ESP32 to SD card |
| SCK | GPIO14 | Clock signal |
| CS | GPIO13 | Chip Select (active low) |

**Assembly of hardware**

1. Preparing the perfboard board
   1. Solder a 2x10 pin header to the perfboard board to connect the TTGO T4
      1. Solder the pin header as high as possible to allow extra room for battery
   2. Ensure that the pins will align to fit the device into the 3D printed container
      1. Use J to S columns on perfboard
2. Wire the MLX90640 Sensor
   1. Power connections:
      1. Solder VDD of the MLX90640 to 3.3V on the prototyping board
      2. Solder GND to a ground rail on the prototyping board
   2. I²C Connections:
      1. Connect SCL from the sensor to IO26 on the TTGO T4
      2. Connect SDA from the sensor to IO33 on the TTGO T4
   3. Pull-up resistors:
      1. Add 4.7 kΩ resistors between SCL and VDD, and SDA and VDD
3. Add the Low-Pass Filter (if needed, i.e. noise in thermal image)
   1. Place a 0.1 µF capacitor between VDD and GND near the sensor
   2. Place a 1.0 µF capacitor between VDD and GND near the sensor
   3. Place a 4.7 µF capacitor near the sensor on the prototyping board for additional stability

**Wiring Schematic:**

|  |  |  |
| --- | --- | --- |
| MLX90640 Pin | TTGO T4 Pin | Function |
| VDD | 3V3 | 3.3V power supply\* |
| GND | GND | Ground |
| SCL | IO26 (SCL) | I2C Clock – use 4.7kΩ for pull-up resistor |
| SDA | IO33 (SDA) | I2C Data – use 4.7kΩ for pull-up resistor |

\*Do **NOT** use the 5V pin of the TTGO T4

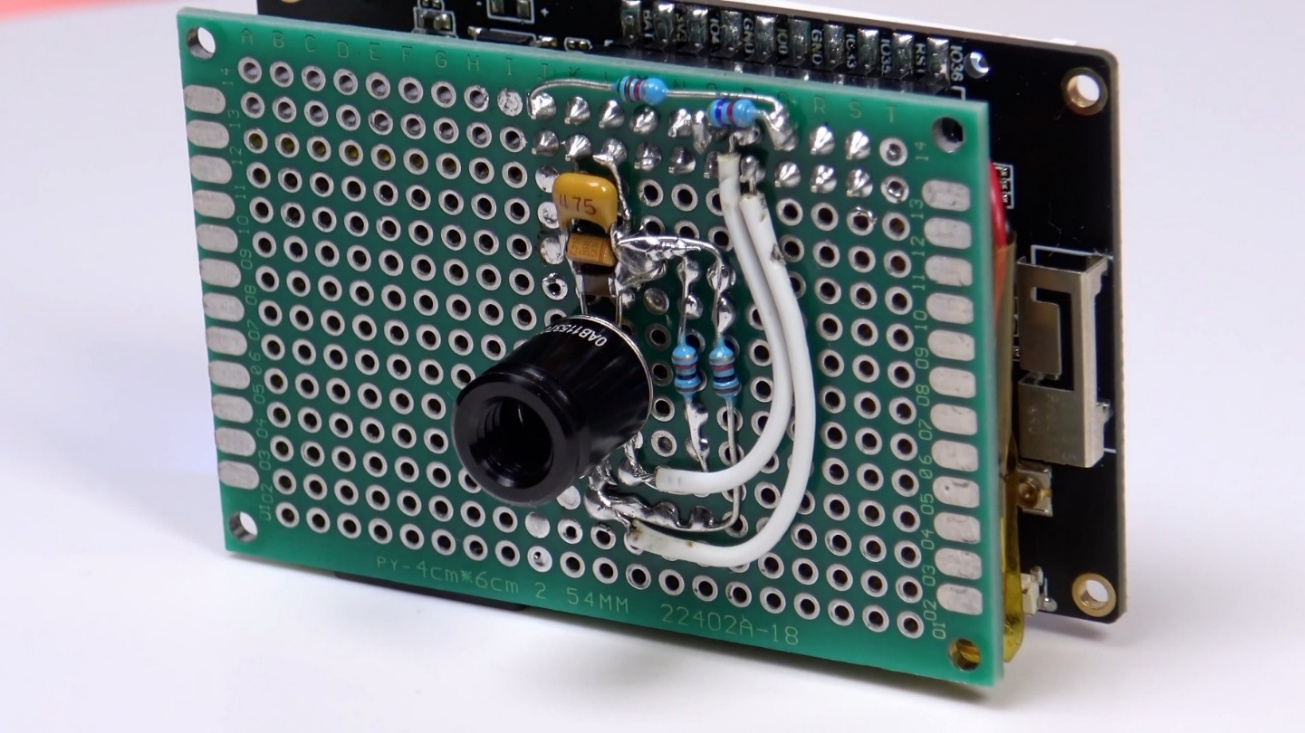


Photo credit: Ruslan Nadyrshin

1. Solder the JST connection to the batter
   1. Seal with shrink wrap to protect wire
2. Test connections
   1. Once the firmware is flashed and the device is powered on, use a multimeter set to 12V to test the GND and VDD connections and pull-ups to ensure the read 3.3V

**Flash the firmware to the TTGO T4**

The creator of this design wrote some amazing firmware that allows for .bmp images to be saved as well as .csv arrays of the thermal data. These are saved to the SD card which can then be opened on your computer. Overall, I am really impressed with the firmware and would use it for future iterations of this device. However, the firmware is written in Russian, so I did write a small python script to translate the Russian to English.

* Translate the Russian to English:
  + Save the python script into the “main” folder of the firmware
  + In bash, cd into the “main” folder of the firmware
  + Run: pip install googletrans==4.0.0-rc1
  + Run: python translate\_russian.py
  + Remove the python file before building the firmware

1. Clone the project’s GitHub repository or download the firmware ZIP for V1.1
   1. <https://github.com/rnadyrshin/mlx90640_thermoimager_v1.1/tree/a2679fc66f4ca5bb4e832fe65e93f84e03494f88>
   2. I had to make a lot of edits to the code to get it to work for my device so ask me for my firmware if you get to this point.
2. Open the ESP IDF bash and cd into firmware directory
3. Run: idf.py build
   1. This will build the firmware for flashing
4. Run idf.py -p (Port) flash
   1. Replace “port” with the port your device is plugged into
5. Run idf.py -p (Port) monitor
   1. This allows you to monitor the device for errors or issues

**Appendix:**

LILYGO T4 specifications

|  |  |
| --- | --- |
| MCU | ESP32 |
| Flash | 4MB |
| PSRAM | 8MB |
| Wireless Connectivity | Wi-fi 802.11 b/g/n, BL V4.2+BLE |
| Programming Platform | Arduino-ide, Micropython |
| Serial chip | CH9102F QFN24 |
| Display Optional | 2. 2 |
| Onboard functions | Buttons: Boot+RST battery power detection |
| ILI9341 TFT LCD | Resolution 320x240 |
| support 65K color display | 4-wire SPI interface |
| Working Power Supply | 3.3V |
| Driver IC | ILI9341 |
| Support arduino library TFT\_ espl | Support USB/LI-PO Battery |
| JST Connect type | JST GH 1. 25mm |