DevNet IoT Produces Portable IR Thermal-Scanner

Leveraging the Cisco IR829 Router to Create a Portable Temperature Screener

Alexander Stevenson

Project GitHub repository: <https://github.com/xanderstevenson/skin-temperature-scanner>

**Who**

----

Hello there, I'm Alexander Stevenson, a Cisco DevNet Engineer and IoT Team member.

I'm based out of Austin, Texas, although our team is mostly divided between

San Jose and Austin, with some members (Florian and Geev), based in Europe and Asia, respectively.

**What**

----

This project, with approval and guidance from managers Paul Zimmerman and Jock Reed, started as a research mission to determine the feasability of deploying IR Thermal cameras on an Edge device and managing them remotely. First, we determined the market lacks hand-held Commercial Off the Shelf (COTS) temperature scanners, except those which are a) battery-operated b) designed for industrial

use (not for body temperature) 3) unable to connect either wired or wirelessly (save for a few models which have Bluetooth connectivity only)

Our initial specificatons included:

1) Portable. The entire sytstem (scanner, router, display and power source) should be able to be transported in a personal carrying case and operated wherever a power outlet and cellular signal are present.

2) Use a Raspberry Pi 4 and an incorporated IT Thermal Camera to collect temperature

3) Extend the range of the thermal data collection by hosting software in the IR829's IOx VM, which will deploy and manage the Thermal Camera and collect dta from it.

4) Create a basic case to house the Raspberry Pi, thermal camera, IR829 and associated peripherals.

**When**

----

This project was started in response to the COVID-19 pandemic in the Spring of 2020.

**Why**

---

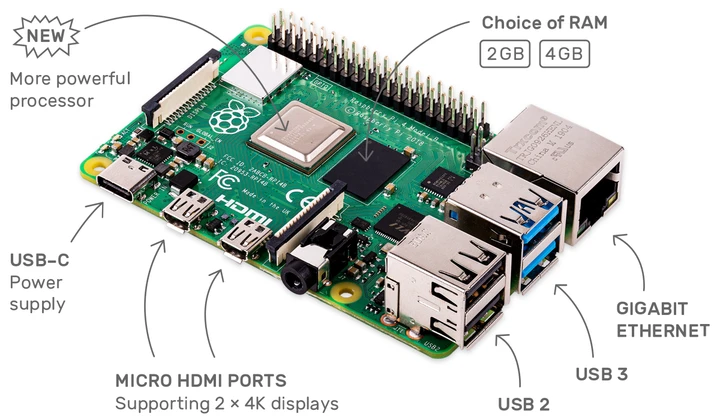
It's economically prohibitive for most people and small businesses to screen people for the main measurable symptom of COVID-19, which is temperature. The current IR Temperature scanners on the market are neither network nor power connectable and are not calibrated to measure human body temperature.

**How**

---

1. Purchase a Raspberry Pi. I purchased a raspberry Pi 4B from Labists, it inlcuded heat-sinks, a case as well as extra wires. Raspian was already preloaded onto the SIM card included in the set.

2. Set up the Pi with a monitor. keyboard and mouse. Harden the Raspberry Pi.



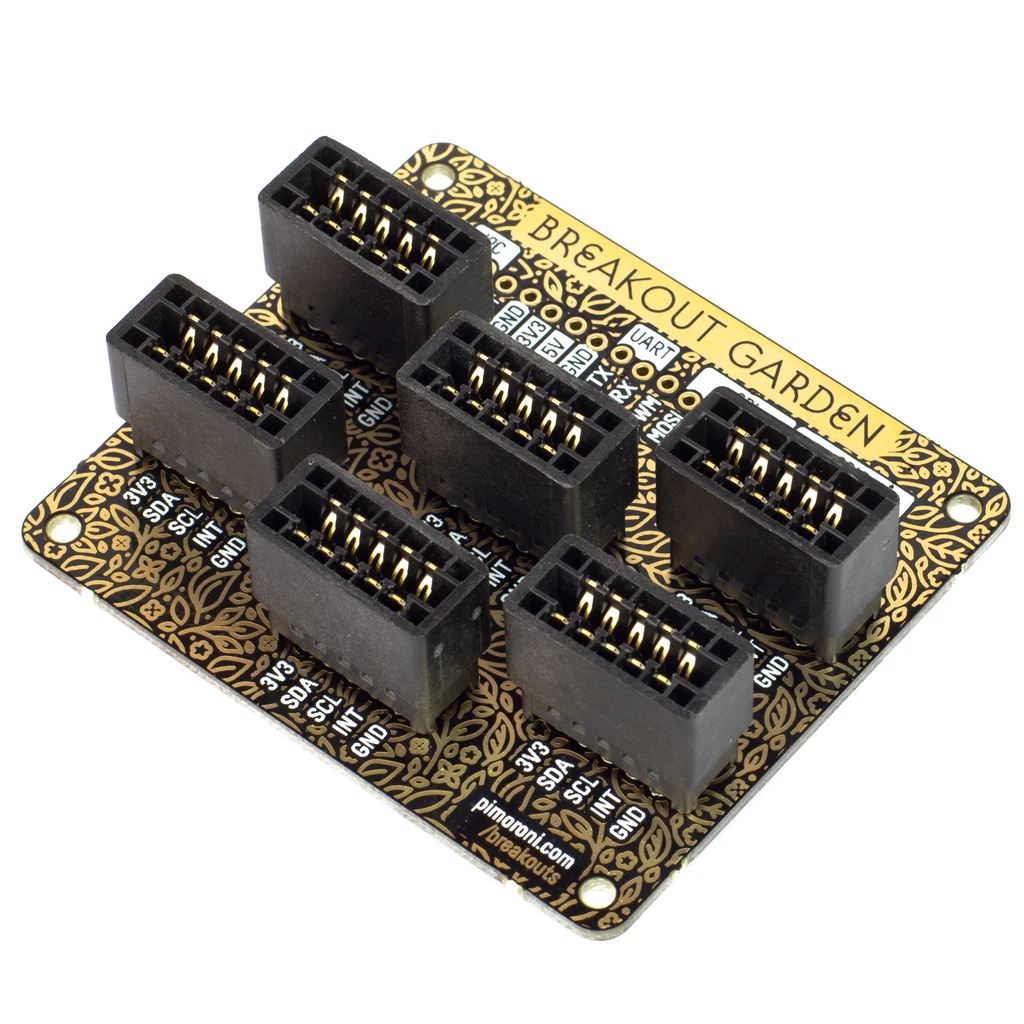
3. Purchase an MLX90640 IR Thermal Sensor. Adafruit is probably the most well-known distributor, but they were all sold out. I ended up ordering mien from E-Bay for around......



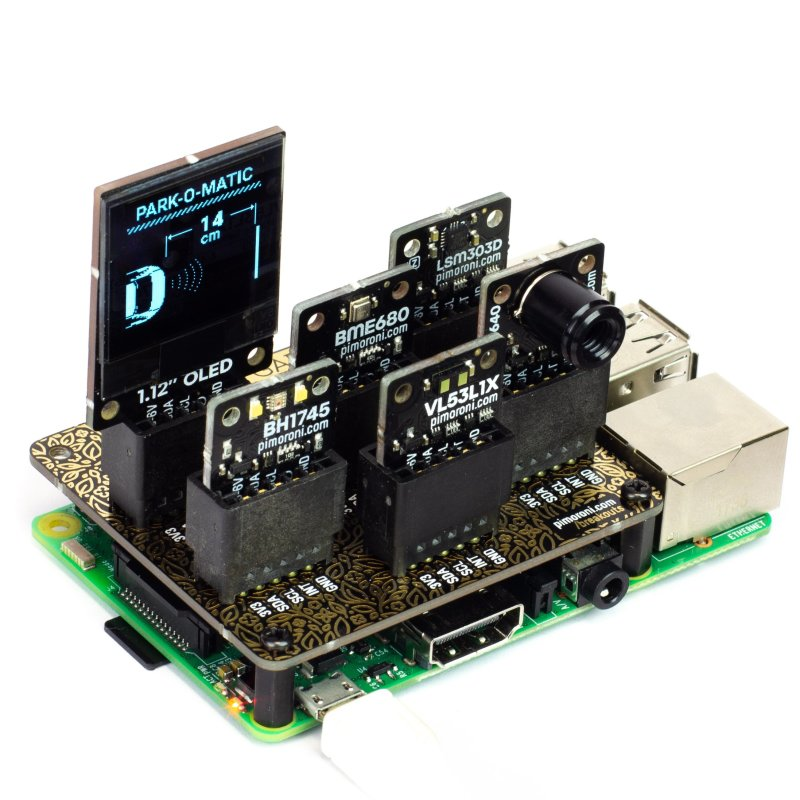
4. Attach the IR Thermal Camera to the Raspberry Pi. There are two ways to do this and each has their advantages.

A) Solder the Camera to the Pi (You are now a computer engineer, yay!), which requires more time and skill but gives you more control over how you want to position the camera in relation to the Pi.

B) Purchase a Breakout Garden for your Raspberry Pi

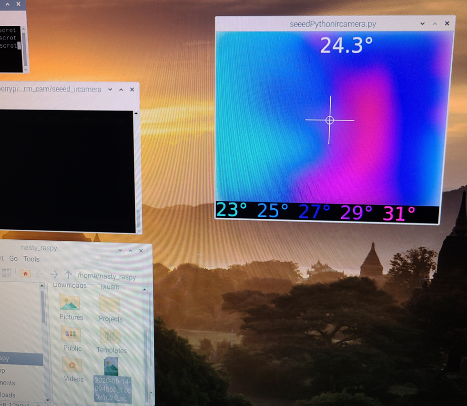
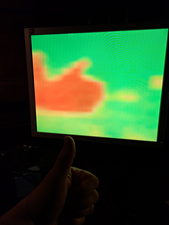
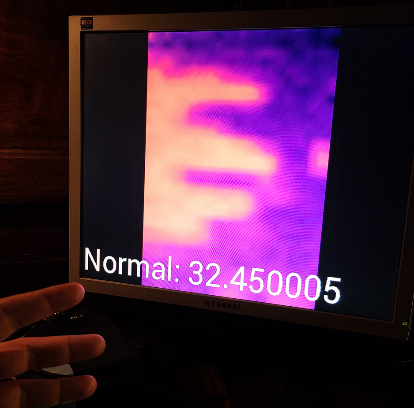


This makes adding the IR Thermal Camera, as well as many other sensors and peripherals rather easy, but does limit the directions you can point your sensors.



Note the IR Thermal Camera in the Breakout Garden, which sits snugly on top of the Pi.

5. Research and test various software, in different languages, to run the IR Thermal Cam and display accurate temperature readings on the monitor.

A handful of engineers around the world have coordinated similar projects involving the Raspberry Pi and an IR Thermal Camera. The images above are from the three finalists in my quest for code which can be applied to our current project.

The best software I found, which performs IR Thermal imaging with instant on-screen color and ordinal representation of an object’s temperature.

Here is the repo: <https://github.com/gilbertfrancois/skin-temperature-scanner>

The engineer who wrote the code, Gilbert François, is a Dutch/French polymath who described themself as such: “an accomplished classical musician and worked as an aerospace scientist, software creator and filmmaker before embracing his true métier in photography.”[[1]](#footnote-1)

All of the code I’ve found online regarding using an IR Thermal Camera on a Raspberry Pi is written in Python, save for one. The goods news is that Python is my best language. The bad news is, the code written by Gilbert François is written in C++.

For those of you new to coding, programming languages are most-likely more similar than you realize. Yes, they can vary greatly in organization, nomencature and readability, but once you learn one of them, you can at least make sense of the other others. I’m able to understand and adapt this C++ code to my needs; that’s not the ~~problem~~ challenge. The challenge will be packaging this code in a Docker container to be deployed on an IR829 Router. I’ve only ever done that with Python, as it’s the lingua franca of IoT. I’m not entirely sure it can be done withh C++, but there’s only one way to find out.

1. (François, n.d.) [↑](#footnote-ref-1)