**Cisco DevNet IoT - Portable IR Thermal-Scanner Project**

**Leveraging the Cisco IR829 Router to Create a Portable Temperature Screener**

Alexander Stevenson

**Who**

Hello, 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, California and Austin, with some members (Florian Pachinger and Geev Cheria), 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 feasibility of deploying an IR Thermal camera on an Edge device and managing it remotely, from the cloud. 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) c) unable to connect either wired or wirelessly (save for a few models which have Bluetooth connectivity only)

Our initial requirements specifications included:

1) Portable. The entire system (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 data 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.

**Where**

The Raspberry Pi, IR Thermal Camera, mouse, monitor, power supply and IR829 Router are all located in my personal IoT Lab in my home.

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

**Why**

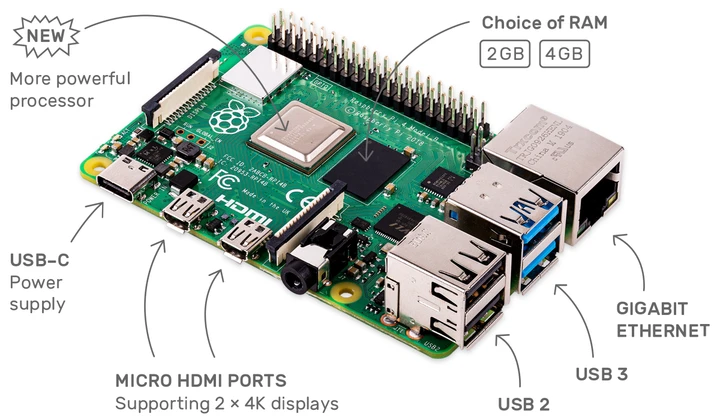
It's economically prohibitive for most people and small businesses to screen people for the main measurable symptom of COVID-19, which is temperature. The current IR Temperature scanners on the market are neither network nor power connectable and are not calibrated to measure human body temperature.

**How**

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

- Raspberry Pi 4B data sheet: <https://www.raspberrypi.org/documentation/hardware/raspberrypi/bcm2711/rpi_DATA_2711_1p0_preliminary.pdf>

2. Set up the Pi with a monitor. keyboard and mouse. Harden the Raspberry Pi.[[1]](#endnote-1)



3. Purchase a MLX90640 IR Thermal Camera. Adafruit and Pimoroni are among the most well-known distributors, but they were all sold out. I ended up ordering a Pimoroni from eBay for around $100 USD, which is roughly double the normal, non-pandemic price. I saw some who tried this project previously had used an 8 X 8 array of IR thermal sensors, which produced a display which lacked detail and granularity. The main reasons I chose the MLX90640 is because it offers a 24x32 array of IR thermal sensors.



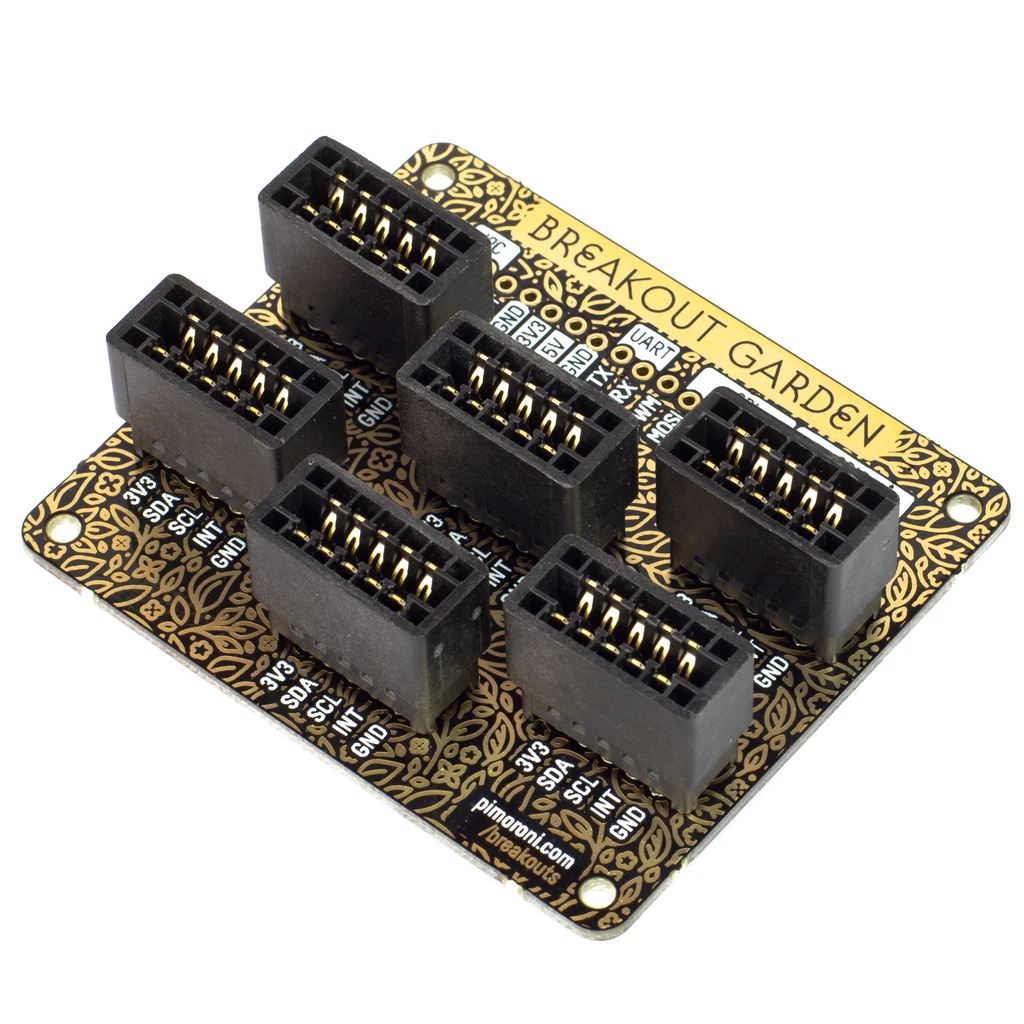
**Specifications of MLX90640 Thermal Camera Breakout – Wide angle (110°)**

* 768 (32x24) thermal sensors
* Detect temperatures from -40 to 300°C
* Approximately 1°C accuracy
* Up to 64 Frames per Second (FPS)
* Wide Angle lens well-suited for room monitoring
* MLX90640 Data Sheet: <https://cdn.sparkfun.com/assets/7/b/f/2/d/MLX90640-Datasheet-Melexis.pdf>
* MLX90640 GitHub repo: <https://github.com/melexis/mlx90640-library>

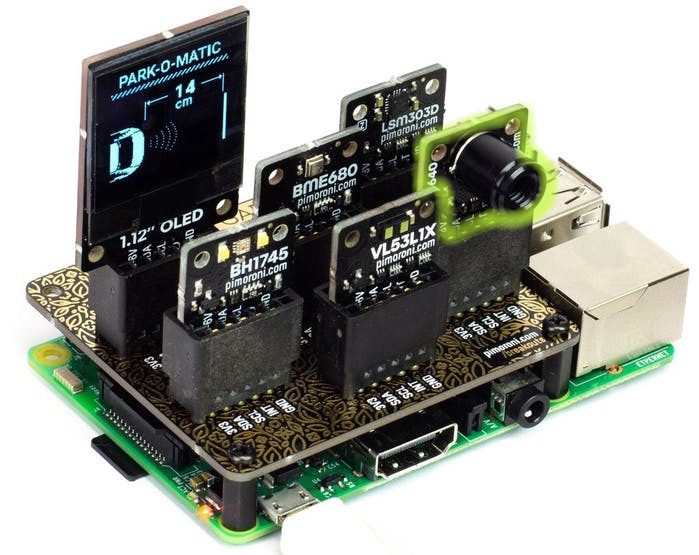
4. Attach the IR Thermal Camera to the Raspberry Pi. There are two ways to do this and each has their advantages:

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

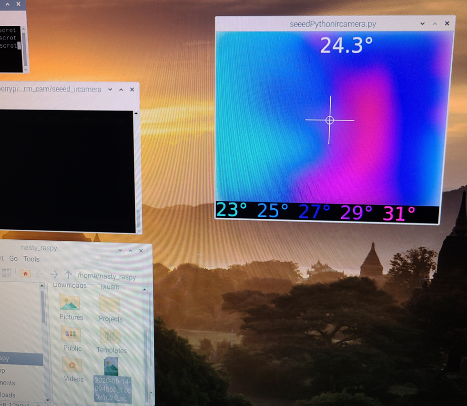
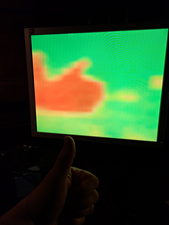
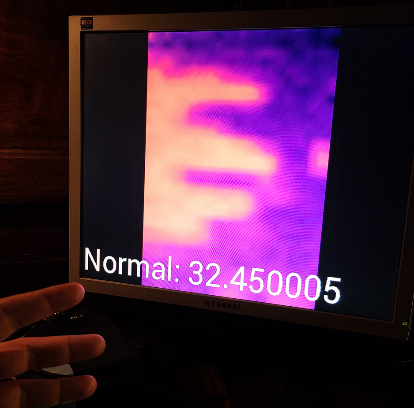


This makes adding the IR Thermal Camera, as well as many other sensors and peripherals rather easy, but does limit the directions you can point your sensors.



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.

winner

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 original 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.”[[2]](#endnote-2)

Another reason I chose this project’s code to incorporate into my project is the fact the programmer, Monseur François, conducted scientific research on thermometers and skin-temperature, in order to make the final product accurate and viable for real-world scenarios.

* Research regarding what temperature normal, healthy human skin should measure found that “the normal temperature of skin is about 33 °C or 91 °F.” The normal range of human skin temperature is found in the range of 32-34 degrees Celsius.[[3]](#endnote-3)
* Research into the emissivity of a material, which is a measure of a material’s radiating efficiency. A score of 1.00 would indicate a material is 100% efficient at radiating energy. According to this research, human skin has an emissivity of 0.99.[[4]](#endnote-4)
* For our project, we then determine the normal range of skin temperatures to be between 32 \* 0.99 and 34 \* 0.99.

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, the best code, 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.

6. Deploy the IR Thermal Camera software to the IR829 Router and physically connect the Raspberry Pi and IT Thermal Camera to the IR829.



* The Cisco IR829 Industrial Integrated Services Router (IR829) is a ruggedized integrated services router designed to be deployed in harsh and industrial environments.
* IR829 Data Sheet: <https://www.cisco.com/c/en/us/products/collateral/routers/829-industrial-router/datasheet-c78-734981.html>
* Versatile. This router is capable of transmitting via:
  + - Multimode 4G LTE
    - 3G wireless WAN (dual active LTE and single LTE models)
    - IEEE 802.11a/b/g/n WLAN (WiFi)
    - Ethernet (RJ45 and SFP)
    - Serial connections and Console port
    - Low Power Wide-Area (LPWA) access using Cisco Interface Module for LoRaWANTM
* Designed to withstand hostile environments including shock, vibration, and humidity, as well as a wide temperature range (-40°C to +60°C and type-tested at +85°C for 16 hours)
* Rapidly deploy a wide variety of Internet of Things (IoT) solutions, including fleet management, mass transit, and remote asset monitoring.
* Includes enterprise-grade wireline-like services such as Quality of Service (QoS), Cisco advanced VPN technologies (DMVPN and Flex VPN) and multi-VRF for WAN, highly secure data, voice, and video communications and Cisco IOx, an open, extensible environment for hosting applications at the network edge

1. (Securing your Raspberry Pi, 2020) [↑](#endnote-ref-1)
2. (François, 2015) [↑](#endnote-ref-2)
3. (Farzana, 2001) [↑](#endnote-ref-3)
4. (Optotherm Support: Emissivity in the Infrared, 2018) [↑](#endnote-ref-4)