A Low-Cost Thermal Imaging Device for Monitoring Electronic Systems Remotely

Jack Humphreys[1] and Emanuele Lindo Secco[1[0000-0002-3269-6749]]

[1] School of Mathematics, Computer Science & Engineering, Liverpool Hope University, L16 9JD, UK
19002075@hope.ac.uk, seccoe@hope.ac.uk

Abstract. This work aims at providing a low-cost solution to produce good quality thermal image that can be monitor remotely. The main application of such a system is to monitor electronic systems or installations as most industrial fires are caused by electronic or electrical systems. The paper presents the design and integration of the system, as well as a set of preliminary validation. The device was developed by combining and integrating a low-cost Raspberry Pi board and Pimoroni MLX90640 thermal camera with C++ and Python programming software. Multiple tests were then performed to validate the system's accuracy. The temperature of a daily life hot object was measured in time and compared vs the thermal measurements of a commercial IR gun (Etekcity Lasergrip 1080). Tests were performed indoor and outdoor to double check the effect of the environmental noise. The overall results show an average difference of +/- 1.9 °C.

Keywords: Electronics, Thermal Imaging, Raspberry Pi, Remote viewing, Servo.

1 Introduction

Currently devices that provide the user with an accurate video feed of temperature and various other forms of data are expensive and not always easily obtainable. There are many different occupations, hobbies and DIY jobs which can require the need to use this information. In this context, this study aims at delivering a low-cost device for automatic temperature detection for use in real-world applications.

Moreover, this work aims to be a very practical solution that will allow the user to connect remotely to the device or connect a display for a more portable solution. Viewing remotely is important if the device is required to be left monitoring an area where humans cannot enter without the proper protection.

However, at a first instance, the focus will be on a system that is stationery and monitors and processes different forms of data to do with thermal imaging.

2 Methodology & Research Design

This section details how the research has been carried out, with details of the design and development process. This design section will convey the hardware and software requirements that were necessary. As well as diagrams planning the layout of the system, describing the choices that were made to reach a good standard.

The methodology details how the development of the project progressed step by step. The prototype system will be described, and the reasoning behind it will be explained. All new code produced is commented and tested properly. Furthermore, the device has been required to be tested in different locations and at differing temperatures to figure out how accurate and useful the data collected is.

2.1 Hardware

The proposed system is made of an embedded computer (namely a Raspberry Pi board), a thermal camera and a servo motor in order to orient the camera (Figure 1).

The main board will be a Raspberry Pi 4B with 2GB of RAM. Other similar boards like the Pi 3 could also be used, or a board will less RAM. However, the Pi 4 currently has the highest processing power and will allow for the generated image to be as good as possible on a Pi.

A starter kit was chosen from ThePiHut. Included in the pack is a: Pi 4, 32GB Class 10 A1 MicroSD, Pi 4 Case, Micro-HDMI Cable, and a USB-C Pi Power Supply. The HDMI cable is not necessary for the final product as the device will be interacted with remotely, however it is needed for the development and debugging phase.

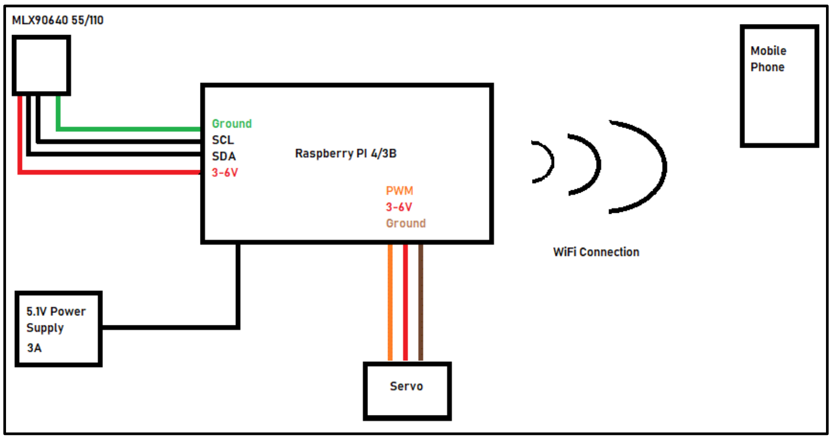

Fig. 1. System Design

There are multiple other versions of the MLX90640 by other developers, but the Pimoroni camera seemed like the best option as the pin layout maps very well to the layout of pins on the Raspberry Pi. Otherwise though, all of them are similar and have an array of 768 thermal sensors which is equal to 32x24 pixels [1]. It is capable of detecting temperatures from -40 to 300 °C with an accuracy of 1 °C. These are good statistics for the price points and should be able to produce the desired image based on projects created by other users. If the device needed to be tailored more towards security and surveillance, then there is a version with a greater field of view at 110. The device used for this project will be the 55 version as there should be slightly better performance for the required application.

A micro servo motor is an optional extra that may or may not be necessary. The servo provides a useful feature in order to be able to re-direct the direction the thermal camera when, for example, observing and monitoring the temperature of an external plant. Although, the idea is to add on screen buttons to the application that allow the user to remotely control the motor so that the camera can turn left and right. If this works successfully then it is a satisfactory proof of concept, and another axis could be added in the future for increased control.

Naturally, the easiest way to remotely view the camera will be with a mobile phone. Any phone that has access to Wi-Fi or a mobile network will most likely be capable of running the remote viewer software.

2.2 Software

The operating system recommended to use with a Raspberry Pi is Pi OS, a Linux based system that works very well for almost any task that can be completed on a normal desktop OS. A few online sources were used to find out the best way to set everything up properly [2, 3].

Pimoroni provides their own python library and examples to allow the programmer to develop their own application [1]. This is essential for making the camera function, however there are other libraries by other developers available. This is heavily adapted from the Melexis library.

Geany is an IDE and text editor that comes pre-installed with Pi OS. Geany is capable of compiling C++ and Python code so can be used for this project. C++ will be used for generating the image, the user interface will be created in Python [4-7].

VNC Viewer [8] is another application that is pre-installed with Pi OS and it that allows the device to be viewed remotely. From prior experience VNC Viewer is very easy to setup and there is very little input delay if the wireless connection is stable so is the obvious choice for this project.

There are also a few other libraries which are needed as recommended by Pimoroni to use the full potential of the camera and some libraries for the python GUI and GPIO control. They are:

- C++: Linux I2C dev library, Bcm2835 library, Linux SDL2 dev library
- Python: Tkinter Python interface library, RPi.GPIO library, time library [7].

2.3 Hardware Design Layout and Test Setup

The MLX90640 is connected to pins numbers 1, 3, 5 and 9 of the Raspberry Pi board. In this order the camera connectors are respectively connected to the 3V3

Power, GPIO 2 (SDA), GPIO 3 (SCL) and Ground pins. Similarly, the wires of the servo are connected to the board, according to the color codes provided by the manufacturer. The same color code is also reported within Figure 1 for convenience. A 1 kΩ resistor is added to the servos Pulse Width Modulation (PWM) wire in order to limit the current from the GPIO12 pin and protect the system in case connection mistakes may occur. The power supply then connects to the USB-C port. Figure 2 shows the overall integrated system, according to the scheme of Figure 1.

Since the Pi 4 has a built-in dual band Wi-Fi (IEEE 802.11), the system is also inherently able to wirelessly connect to other devices such as, for example, mobile phones; similarly, a desktop PC can be remotely streamed without the use of any physical connections such as a HDMI cable as portrayed in Figure 1.

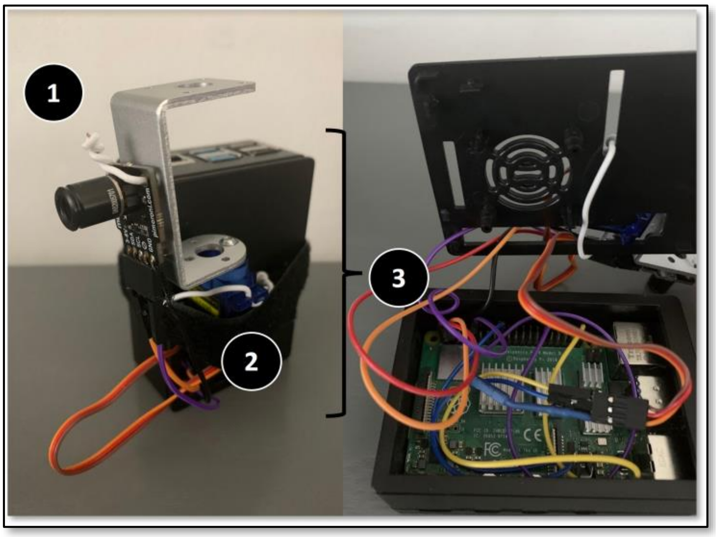

Fig. 2. System integration combining the low-cost MLX90640 thermal camera (1) and servo-motor (2) with the Raspberry Pi and Wi-Fi communication module (3).

2.4 Testing Procedure

The first test will be completed by measuring the temperature of a cup of coffee over the period of 2 min with a sampling rate of 10 s. Figure 3 shows the experimental set-up. The results of this test are reported in Table 1. This will be useful to prove whether the device is accurate or not. Hopefully, the test will be long enough to produce a substantial enough difference every ten seconds so that a range of temperatures can be tested. If the difference is negligible then the time limit can simply be extended, and more readings can be taken. It is worth noting that the MLX90640 datasheet claims to have an accuracy of ± 1°C, whilst the IR gun has an accuracy of ± 2°C so it will not be a surprise should there be a small difference within that range.

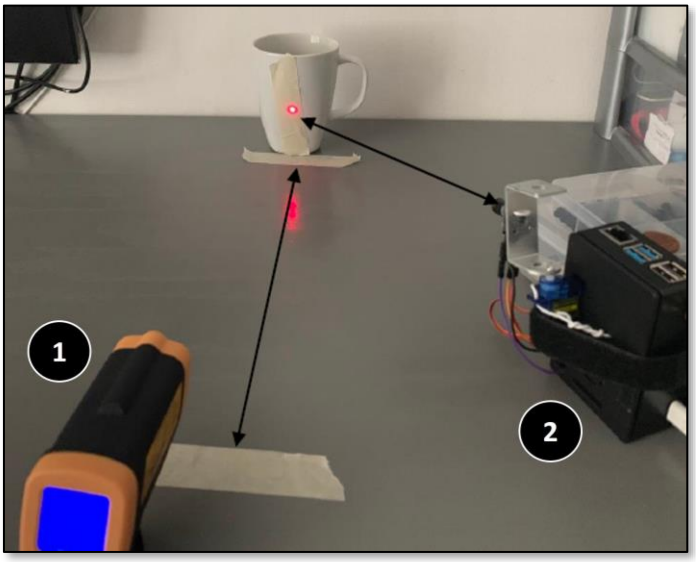

Fig. 3. Experimental setting of the coffee cup indoor test with the Thermal IR Gun, Etekcity Lasergrip 1080 (1) and the low-cost Raspberry Pi - based system with the MLX90640 Thermal Camera (2).

The test will take place twice, with one done outside and the other inside. Therefore, a temperature difference column is important to remove any initial difference between both tests allowing it to be as fair as possible. If the device is working correctly then both tables should follow a similar pattern in relation to the indoor and outdoor environmental temperature difference.

Table 1. Test 1 results.

t[s]	Temperature [°C]		
	MLX90640	Etekcity Lasergrip 1080	Δ
0	47.5	49.7	2.2
10	47.3	49.5	2.2
20	47.1	49.3	2.2
30	46.9	49.1	2.2
40	46.9	49.1	2.2
50	46.8	49.1	2.3
60	46.9	49.0	2.1
70	46.7	48.9	2.2
80	46.6	48.8	2.2
90	46.7	48.8	2.1

100	46.2	48.7	2.5
110	46.4	48.6	2.2
120	46.3	48.4	2.1
130	46.5	48.4	1.9
140	46.2	48.3	2.1
150	46.1	48.1	2.0
160	45.9	47.9	2.0
170	45.8	47.8	2.0
180	45.7	47.5	2.2

Ideally, there will be no real difference between the two tests apart from the rate at which the temperature decreases.

3 Results & Discussion

The results of the first successful temperature test have been shown in Figure 4. The temperatures recorded by the Raspberry Pi can be seen along the red line and the temperatures by the IR gun are indicated by the blue line. The ambient temperature was 23.5 °C at the time, this will be useful when used in comparison with the results from the second temperature test.

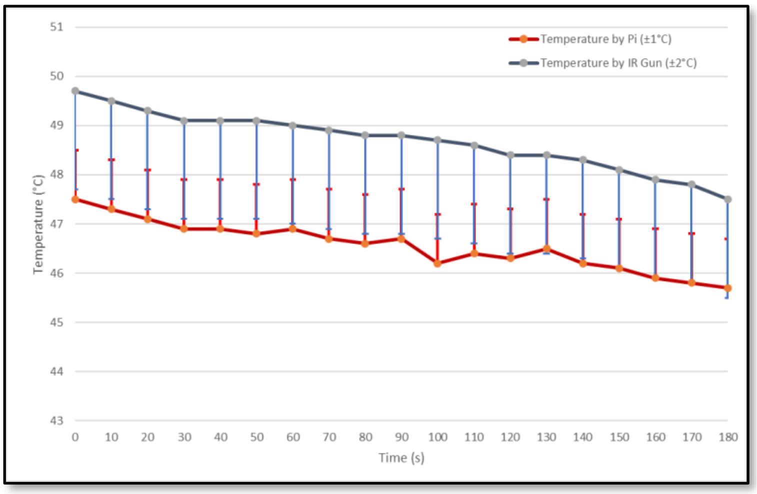

Fig. 4. Results of Test 1.

Furthermore, from the data collected it can be assumed that the device developed on the Raspberry Pi is functioning properly under the test conditions There is a constant gap between both devices of at least 1.9 °C. Whilst this seems like quite a large margin, it is not a failure as once the margin of error is taken into account then the temperatures will always overlap as displayed in Figure 4. Both devices come with a margin of error provided by the manufacturer which is useful for determining whether the results are accurate or not. Since the temperature difference is so consistent this seems like the most likely cause of the gap. However, multiple times the temperature detected by the Pi appeared to have numerous drops and increases in temperature whereas the IR gun only dropped in temperature throughout the test. The first example of this appears at around 50 s when the temperature drops to 46.8 °C, but then increases 10 s later to 46.9 °C. This occurred four times during the test with the worst example being a 0.2 °C increase. Obviously, this is not a major problem as the difference is not greater than 1 °C and is probably caused due to the large amount of data that the camera detects in comparison the IR gun. Constantly detecting a 32x24 area will result in some anomalies or missed readings. Furthermore, even though the code provided by Pimoroni adapts in case of dead pixels this could be a potential cause of the problem.

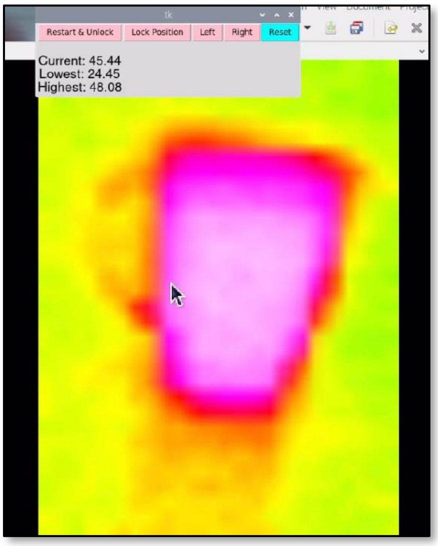

Fig. 5. The outcome of the GUI and SDLSCALE Program which allow the visualization of the thermal image.

The second test took place with all the same conditions, the only difference being the environment that the results were recorded in. The ambient temperature at the time was around 13 °C, which was a 10.5 °C difference to the previous test. This is evident in the results which behaved relatively the same to test one, and as expected the rate of temperature change increased. However, the average temperature difference between the two devices also increased.

The average of the first test was a difference of about 2.15 °C, whereas the second average taken was around 2.27 °C. This is not too worrying as they are not vastly different but could become a problem if used in harsher conditions. Although, the problem may not lie solely with the Pi and could be caused by the IR gun. Overall, both tests were successful and showcased the effectiveness of the device as the readings were within an acceptable level. With further refining and understanding of the software the problems shown could be mitigated with future development.

Figure 5 displays a successful implementation of the objectives as the temperatures can clearly be seen by the user and the motor moves when the buttons are pressed. Figure 5 showcases what the configuration of the Raspberry Pi system looks like up closely. This thermal image has been produced by following the instructions provided with the Pimoroni MLX90640 library which was adapted from the Melexis MLX90640 library. The example program SDLSCALE has been modified to allow for the data to be exported for the creation of the GUI.

4 Conclusion

The main purpose of this work was to create a device that could produce an accurate thermal image that could be viewed remotely by the user. This condition has been met as highlighted in the results section. Additionally, the servo motor was also implemented successfully to allow the user to control the direction that the camera is facing, which can also be done remotely. Ultimately, all the aims and objectives of this project have been completed. Further research should be implemented vs the processing and analysis of the image in order, for example, to automatically detect any hazard condition and extract useful information [9-11]. An in-depth validation of the system in real industrial scenario is missing and would be of benefit for a proper testing of the proposed architecture. Computer vision skills could also be implemented in a further stage where the system automatically recognize and classify the information inherently reported on the image of the thermal measurement [12-15]. Moreover, the wireless communication system and reliability should be tested as well, providing a final overview of the system performance.

Acknowledgment

This work was presented in dissertation form in fulfilment of the requirements for the BEng in Electronic and Computer Engineering for the student J Humphreys under the supervision of EL Secco from the Robotics Laboratory, School of Mathematics, Computer Science and Engineering, Liverpool Hope University.

References

1. Pimoroni. 2022. Python library for the MLX90640 thermal camera. [online] Available at: <https://github.com/pimoroni/mlx90640-library> [Accessed 31 March 2022].
2. Hrisko, J., 2020. High Resolution Thermal Camera with Raspberry Pi and MLX90640. Maker Portal. [online] Maker Portal. Available at:

<https://makersportal.com/blog/2020/6/8/high-resolution-thermal-camera-with-raspberry-pi-and-mlx90640> [Accessed 13 April 2022].

3. Shaffner, T., 2021. MLX90640 Thermal Camera with Raspberry Pi 4. [online] PiThermalCam. Available at: <https://tomshaffner.github.io/PiThermalCam/> [Accessed 13 April 2022].
4. Docs.python.org. 2022. tkinter — Python interface to Tcl/Tk — Python 3.10.4 documentation. [online] Available at: <https://docs.python.org/3/library/tkinter.html> [Accessed 13 April 2022].
5. 2022. Raspberry Pi OS – Raspberry Pi. [online] Raspberry Pi. Available at: <https://www.raspberrypi.com/software/> [Accessed 13 April 2022].
6. Melexis. 2021. MLX90640 library functions. [online] Available at: <https://github.com/melexis/mlx90640-library> [Accessed 13 April 2022].
7. PyPI. 2022. RPi.GPIO. [online] Available at: <https://pypi.org/project/RPi.GPIO/> [Accessed 13 April 2022].
8. RealVNC®. 2022. *RealVNC® - Remote access software for desktop and mobile | RealVNC*. [online] Available at: <https://www.realvnc.com/en/> [Accessed 28 June 2022].
9. EL Secco, R Abdulrahman, I Felmeri, AK Nagar, Development of Cost-effective Endurance Test Rig with Integrated Algorithm for Safety, Soft Computing for Problem Solving, Advances in Intelligent Systems and Computing, 1139, chapter 14, Springer, DOI: 10.1007/978-981-15-3287-0_14
10. D McHugh, N Buckley, EL Secco, A low-cost visual sensor for gesture recognition via AI CNNS, Intelligent Systems Conference (IntelliSys) 2020, Amsterdam, The Netherlands
11. EL Secco, DD McHugh, N Buckley, A CNN-based Computer Vision Interface for Prosthetics’ application, EAI MobiHealth 2021 - 10th EAI International Conference on Wireless Mobile Communication and Healthcare, 41-59
12. Buckley N, Sherrett L, Secco EL, A CNN sign language recognition system with single & double-handed gestures, IEEE Signature Conference on Computers, Software, and Applications, 2021
13. K. Myers, E.L. Secco, A Low-Cost Embedded Computer Vision System for the Classification of Recyclable Objects, Congress on Intelligent Systems (CIS - 2020), Intelligent Learning for Computer Vision, Lecture Notes on Data Engineering and Communications Technologies 61
14. D. McHugh, N. Buckley, E.L. Secco, A low-cost visual sensor for gesture recognition via AI CNNS, Intelligent Systems Conference (IntelliSys) 2020, Amsterdam, The Netherlands
15. A.T. Maereg, Y. Lou, E.L. Secco, R. King, Hand Gesture Recognition Based on Near-Infrared Sensing Wristband, Proceedings of the 15th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2020), 110-117, 2020