T.C.
DOKUZ EYLÜL UNIVERSITY
ENGINEERING FACULTY
ELECTRICAL & ELECTRONICS ENGINEERING
DEPARTMENT

Smart Mirror Controller

Final Project

by

Rabia DOĞAN

Advisor

Ph.D. Özgür TAMER

January,2021
IZMIR

THESIS EVALUATION FORM

We certify that we have read this thesis and that in our opinion it is fully adequate, in scope and qualify as an undergraduate thesis, based on the result of the oral examination taken place on —/—/———

Ph.D. Özgür TAMER
(ADVISOR)

Prof. Dr. Gülay TOHUMOĞLU
(COMMITTEE MEMBER)

Ph.D. Abdül BALIKCI
(COMMITTEE MEMBER)

Prof. Dr. Mehmet KUNTALP
(CHAIRPERSON)

ABSTRACT

ÖZET

Contents

List of Tables

List of Figures

1. INTRODUCTION

2. TECHNICAL BACKGROUND

2.1 Introduction to Artificial Neural Networks

2.2 Learning Process of Artificial Neural Networks

3. MATERIALS AND METHODS

3.1 Dataset

3.2 Data Augmentation

3.3 Model Architecture for LE-NET5

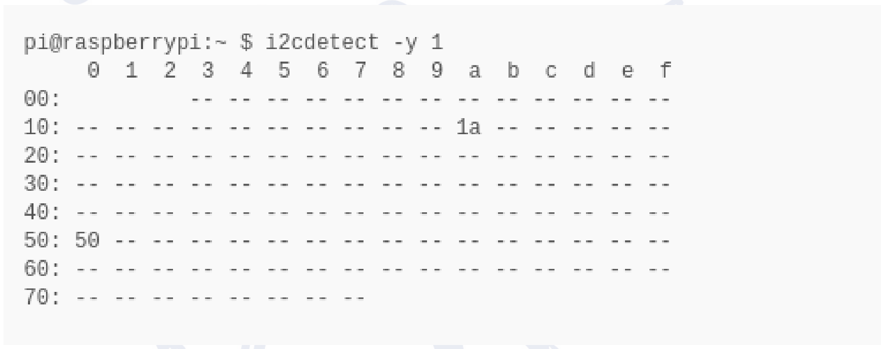
4. PROGRESS AND RESULT

4.1 Capture Thermopiles Image

4.1.1 Heimann Thermopile Array Sensor communication

First, I provided the connections between the sensor and the Raspberry Pi in order to receive the image. I provided the communication with the mini card I made in 4.2 using the I2C protocol.

With the device connected to a Raspberry Pi, and with the Pi configured. [1] correctly for I2C, I was able to see the devices connected with the i2cdetect command.



```
pi@raspberrypi:~ $ i2cdetect -y 1
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- 1a -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: 50 -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
```

Figure 4.1: Thermopile Infrared Array Device and EEPROM Addresses

In order to be able to read the data properly, the python-periphery [3] library was used. The sensor is divided into two parts (Top and Bottom Half), which are also divided into 4 blocks. The reading order is shown below for different blocks. When a conversion is initiated, the X Block of the upper and lower half are measured simultaneously. Each block consists of 128 Pixels sampled entirely in parallel. The reading order in the lower half is mirrored compared to the upper half so the center lines are always read last.

Table 4.1: Read Data 1 Command (Top Half of Array)

Addr/CMD	0x1A (7 Bit!) / 0x0A							
Read Data	7	6	5	4	3	2	1	0
1. Byte / 2. Byte	PTAT 1 MSB / LSB or Vdd 1 MSB / LSB							
3. Byte / 4. Byte	Pixel (0+BLOCK*128) MSB / LSB							
5. Byte / 6. Byte	Pixel (1+BLOCK*128) MSB / LSB							
...								
257. Byte / 258. Byte	Pixel (127+BLOCK*128) MSB / LSB							

Table 4.2: Read Data 2 Command (Bottom Half of Array)

Addr/CMD	0x1A (7 Bit!) / 0x0B							
Read Data	7	6	5	4	3	2	1	0
1. Byte / 2. Byte	PTAT 2 MSB / LSB or Vdd 2 MSB / LSB							
3. Byte / 4. Byte	Pixel (992-BLOCK*128) MSB / LSB							
5. Byte / 6. Byte	Pixel (993-BLOCK*128) MSB / LSB							
...								
65. Byte / 66. Byte	Pixel (1023-BLOCK*128) MSB / LSB							
65. Byte / 66. Byte	Pixel (1023-BLOCK*128) MSB / LSB							
67. Byte / 68. Byte	Pixel (960-BLOCK*128) MSB / LSB							
69. Byte / 70. Byte	Pixel (961-BLOCK*128) MSB / LSB							
...								
129. Byte / 130. Byte	Pixel (991-BLOCK*128) MSB / LSB							
131. Byte / 132. Byte	Pixel (928-BLOCK*128) MSB / LSB							
...								
257. Byte / 258. Bytes	Pixel (927-BLOCK*128) MSB / LSB							

Each block is checked before it is read. The python-opencv [2] library was used to visualize the obtained result.

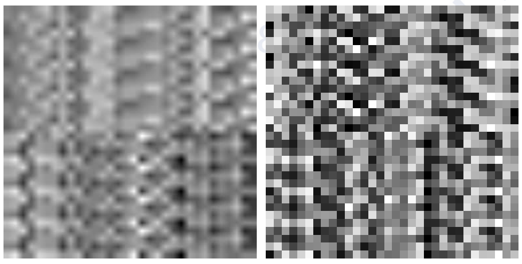


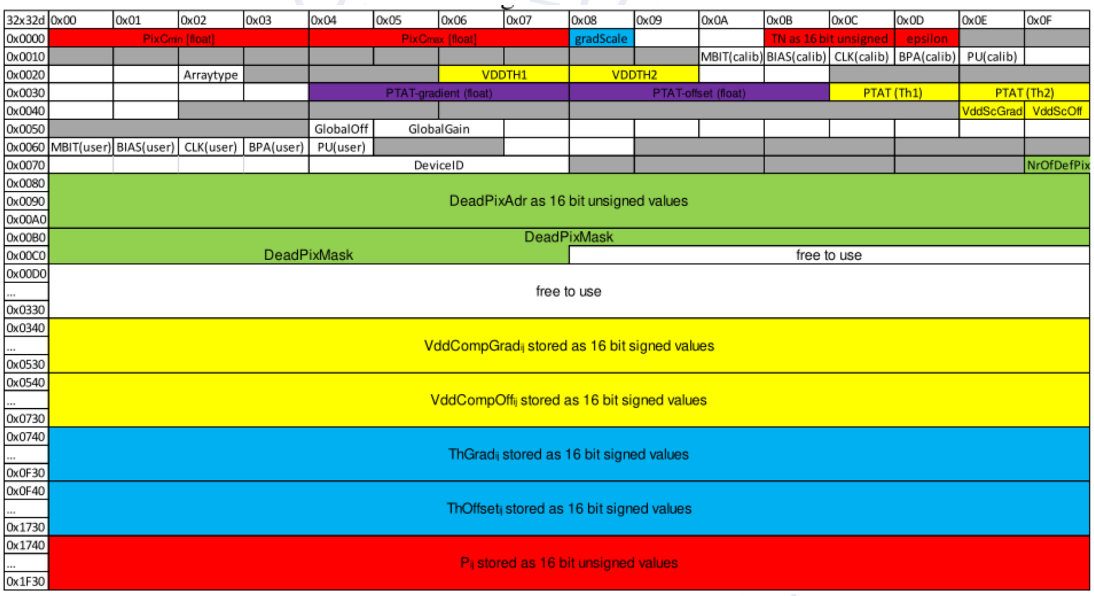
Figure 4.2: Thermal Image

Each pixel (or each analog-to-digital converter, given the repeating structure corresponding to each "block" of the sensor) has its own offset and sensitivity to incident light. Without calibrating it, this constant "noise" suppresses the signal from changing IR/temperature conditions. By subtracting the two frames in quick succession, this common noise signal is removed.

However, it is still quite noisy, as this frame subtraction increases random noise (since we now have contributions from two frames) and does not correct pixel-dependent sensitivity. Only fabrication calibration will be done with EEPROM data in the next step.

4.1.2 Calibrating images from Heimann Thermopile Array Sensor

After reading an image off a Heimann thermopile array, the pixel values can be converted to temperature readings through the use of calibration parameters stored on the device. To extract the calibration parameters, it is easiest to first read off the entire EEPROM on the thermopile array.



32x32d	0x00	0x01	0x02	0x03	0x04	0x05	0x06	0x07	0x08	0x09	0x0A	0x0B	0x0C	0x0D	0x0E	0x0F
0x0000	PixCmin [float]				PixCmax [float]				gradScale			TN as 16 bit unsigned		epsilon		
0x0010											MBIT(calib)	BIAS(calib)	CLK(calib)	BPA(calib)	PU(calib)	
0x0020			Arraytype				VDDTH1		VDDTH2							
0x0030					PTAT-gradient (float)				PTAT-offset (float)				PTAT (Th1)		PTAT (Th2)	
0x0040															VddScGrad	VddScOff
0x0050					GlobalOff	GlobalGain										
0x0060	MBIT(user)	BIAS(user)	CLK(user)	BPA(user)	PU(user)											
0x0070					DeviceID											NrOfDefPix
0x0080	DeadPixAdr as 16 bit unsigned values															
0x0090																
0x00A0																
0x00B0	DeadPixMask															
0x00C0									free to use							
0x00D0 ... 0x0330	free to use															
0x0340 ... 0x0530	VddCompGrad$_{ij}$ stored as 16 bit signed values															
0x0540 ... 0x0730	VddCompOff$_{ij}$ stored as 16 bit signed values															
0x0740 ... 0x0F30	ThGrad$_{ij}$ stored as 16 bit signed values															
0x0F40 ... 0x1730	ThOffset$_{ij}$ stored as 16 bit signed values															
0x1740 ... 0x1F30	P$_{ij}$ stored as 16 bit unsigned values															

Figure 4.3: EEPROM overview 32x32d [4]

Then, parameters and calibration values can be extracted from this array, as described in the Heimann datasheet. [4]

Calibration for only one pixel is done as follows.

$PTAT_{av} = \frac{\sum\limits_{i=0}^{7} PTAT_i}{8} = 38152 Digits$

$PTAT_{gradient} = 0.0211 dK/Digit$ and $PTAT_{offset} = 2195.0 dK$

$V_{00} = 34435 Digits$

$elOffset[0] = 34240$

$gradScale = 24$

$ThGrad_{00} = 11137$

$ThOffset_{00} = 65506$

$VDD_{av} = 35000$

$VDD_{TH1} = 33942$

$VDD_{TH2} = 36942$

$PTAT_{TH1} = 30000$

$PTAT_{TH_2} = 42000$

$VddCompGrad[0] = 10356$

$VddCompOff[0] = 51390$

$VddScGrad = 16$

$VddScOff = 23$

$PixC_{00} = 1 \cdot 087 \cdot 10^8$

$PCSCALEVAL = 1 \cdot 10^8$

Calculation of ambient temperature:

$T_a = PTAT_{av} \cdot PTAT_{gradient} + PTAT_{offset} = 38152 \cdot 0.0211 + 2195.0dK = 3000dK$

Compensation of thermal offset:

$V_{00\_Comp} = V_{00} - \frac{Th_{Grad00} \cdot T_a}{2^{gradScale}} - Th_{Offset_{00}} = 34439$

Compensation of electrical offset:

$V^*_{00\_Comp} = V_{00\_Comp} - elOffset[0] = 199$

Compensation of supply voltage:

$V_{00-VDD_{Comp}} = V^*_{00\_Comp} - \frac{\frac{Vdd_{CompGrad}[0] \cdot PTAT_{av}}{2^{Vdd_{ScGrad}}} + V_{Vdd_{Compoff}}[0]}{2^{Vdd_{ScOff}}}$

$\cdot (VDD_{av} - VDD_{TH1} - (\frac{VDD_{TH2} - VDD_{TH1}}{PTAT_{TH2} - PTAT_{TH1}}) \cdot (PTAT_{av} - PTAT_{TH1})) = 199 - 1 = 198$

The sensitivity coefficients (PixC ij) are calculated:

$PixC_{00} = (\frac{P_{00} \cdot (PixC_{Max} - PixC_{Min})}{65535} + PixC_{Min}) \cdot \frac{epsilon}{100} \cdot \frac{GlobalGain}{100000} = 1 \cdot 087 \cdot 10^8$

Leading to a compensation of the pixel voltage:

$V_{00PixC} = \frac{V_{00-VDD_{Comp}} \cdot PCSCALEVAL}{PixC_0} = 182$

All operations are applied for 1024 pixels. Application result images are as in figure 4.4.

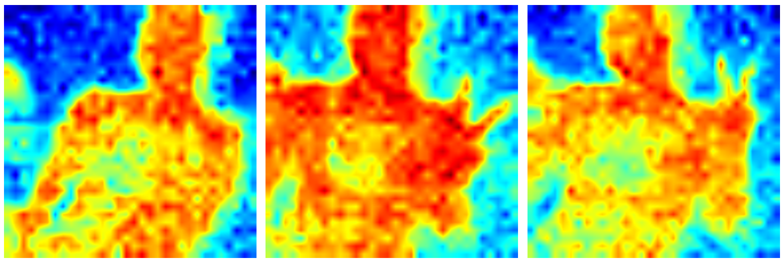


Figure 4.4: Thermal Images with EEPROM Calibration Data

NOTE:All steps to acquired the image are made with reference to the datasheet [4]**.**

4.2 Hand Thermal Image Isolation

The hand was isolated from the background without using any image processing method. For this, it has been arranged in a way that can remove the ambient temperature of the device from the image before giving a command. First, the average of 10 images was taken and given to all images. Thus, the background temperature was isolated.

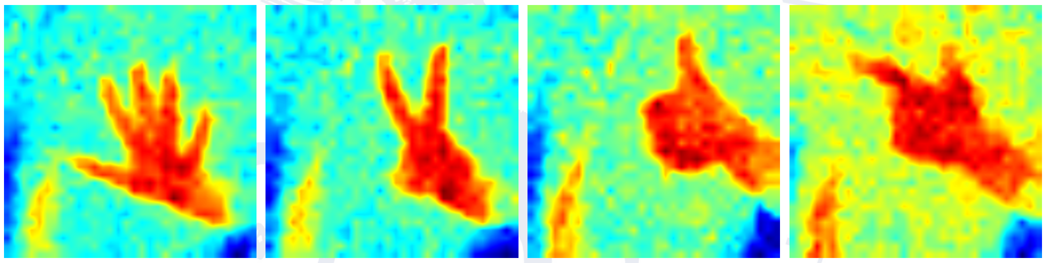


Figure 4.5: Thermal Images with Background

4.3 Hand Gesture Recognition

4.3.1 Static Gesture Recognition

4.3.2 Dynamic Gesture Recognition

4.4 Sensor and MCU Communication PCB Design

A sensor PCB design will be made with the selection and supply of the passive infrared sensor. In the PCB design in question, a schematic design will be made first and the necessary components and connections will be made schematically. After this part, the schematic design will be turned into a printed circuit. In the design of the printed circuit board, along with the schematic design, the physical properties of the place where the sensor will be mounted will be taken into consideration in the current smart mirror design. After this part, the designed sensor PCB will be produced in an external company and its assembly will be carried out in the company.

The pcb and schematic drawings of the communication PCB are as shown in the figures below. Schematic and PCB design will be done using KiCad EDA - Schematic Capture & PCB Design Software. In the next step, support will be received for printed circuit operations. After soldering the printed cards, our sensor communication PCB will be ready and the test phase will begin.

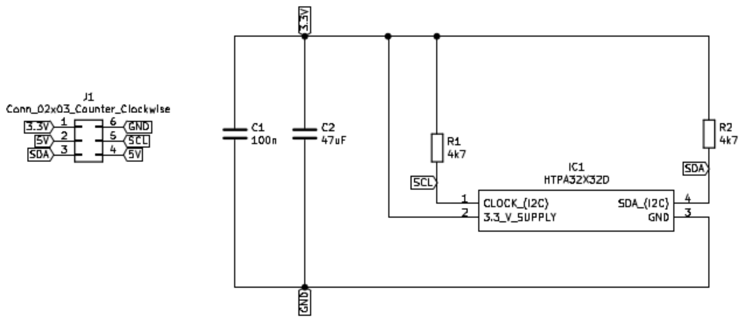


Figure 4.6: Schematic Design of Communication PCB

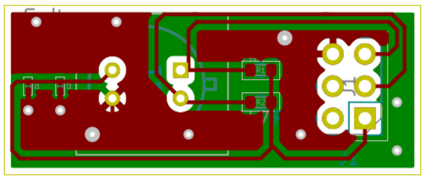


Figure 4.7: PCB Design of Communication PCB

4.5 Matching Gesture to Commands of Smart Mirror

5. COST ANALYSIS

6. CONCLUSION

Bibliography

[1] “Adafruit’s Raspberry Pi Lesson 4. GPIO Setup”.
URL `https://learn.adafruit.com/adafruits-raspberry-pi-lesson-4-gpio-setup/configu`

[2] “OpenCV: OpenCV-Python Tutorials”.
URL $https : //docs.opencv.org/4.5.2/d6/d00/tutorial_py_root.html$

[3] vsergeev. “python-periphery: A pure Python 2/3 library for peripheral I/O (GPIO, LED, PWM, SPI, I2C, MMIO, Serial) in Linux.”
URL `https://github.com/vsergeev/python-periphery`

[4] Lupp, S. “Htpa32x32dr2l5.0/0.85f7.7ehic thermopile array with lens optics rev3.0”. Technical report, Heimann, 2018.03.28.

7. APPENDIX