Netaji Subhas Institute of Technology

EC-316: Microprocessors Lab Project Report

Shreya Seth : 160/EC/15
Sourav Bhattacharjee : 171/EC/15

May 2018

Contents

1 Introduction **3**

2 Detailed Description **4**
2.1 Hardware 4
2.1.1 The Sensor Module 5
2.1.2 The Motherboard 10
2.1.3 Designing and Fabrication of boards 19
2.2 Software 20
2.2.1 Assembly Code of the main algorithm 21
2.2.2 Explanation and pseudo-code for the assembly code . . . 23

3 Results **24**
3.1 Note-wise final output 24
3.2 Final Working Project 24

4 Limitations of the project **26**

Note Detecting Wallet : 8085 Microprocessor based Project Report

Shreya Seth
160/EC/15

Sourav Bhattacharjee
171/EC/15

May 2018

1 Introduction

Paper currency identification is a widely researched field, primarily because of its application in numerous fields, viz. Automated Teller Machines (ATMs), ticket vending machines, food dispensers, counterfeit currency detection and aiding the blind to correctly identify notes of different denominations. These applications demand the need to develop an efficient banknote detection system which can discern between notes in any form, new or used, and from any section of the note, by exploiting one or more unique feature of the currency notes.

Some commonly used techniques for note detection include optical sensing, in which the unique pixel pattern of the note is looked for, proximity detection, which exploits the magnetic properties of the ink used in notes to detect them, and detection based on physical parameters like geometry, colour etc.

This project focuses on detection of Indian currency based on the physical colour of the notes, using Intel 8085 microprocessor and TCS3200 colour sensor as its heart.This project was undertaken as a part of EC-316 Microprocessor course under the guidance of Professor D.V. Gadre. Our motivation to undertake this project was not as noble as helping the blind, but simply to build a handy wallet which helps people keep a track of their expenses by automatically detecting the notes which are being added/removed to/from the wallet.

2 Detailed Description

Any project based on embedded systems will have two interdependent segments: Hardware and Software. Below, we have taken up hardware first.

2.1 Hardware

A block diagram representation of the project hardware is given below. Each block has been taken up and described in the subsequent sections.

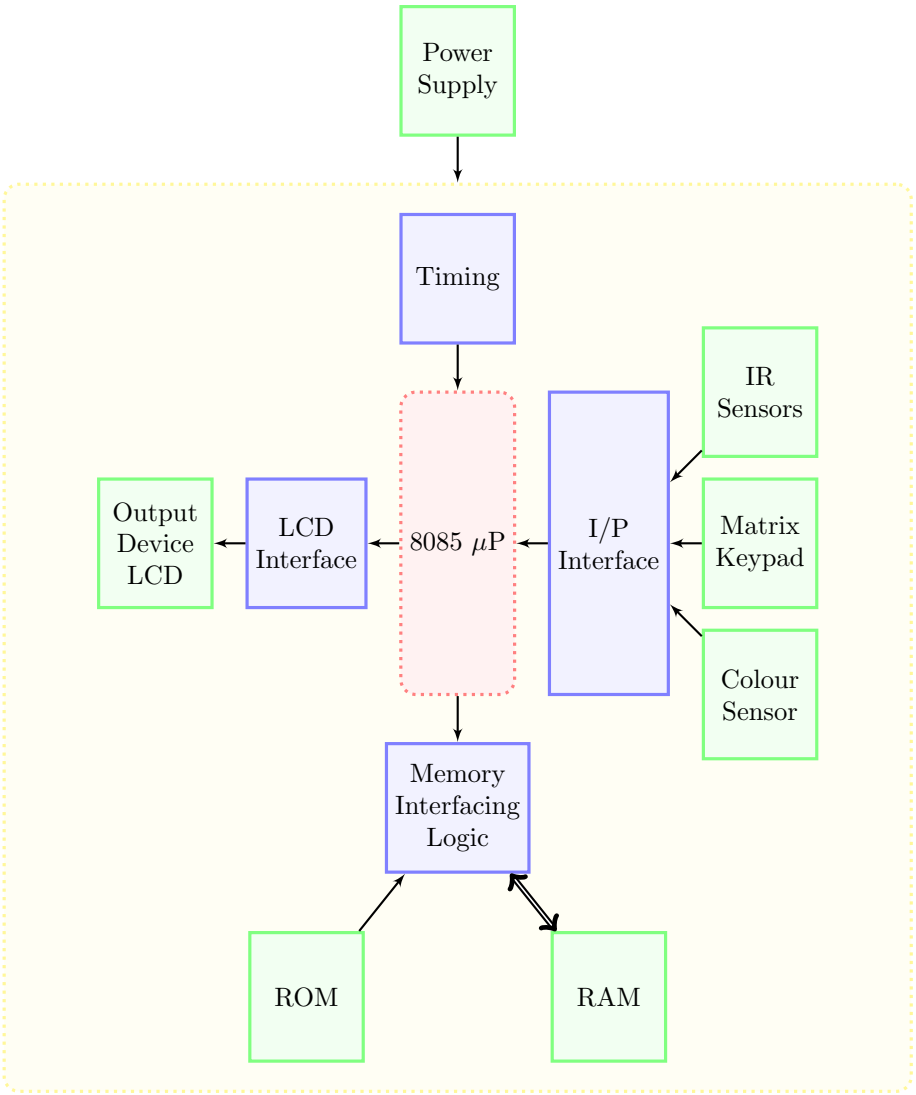


Figure 1: Block Diagram Representation of the Hardware

The project hardware has two parts to it: the motherboard and the sensor module.

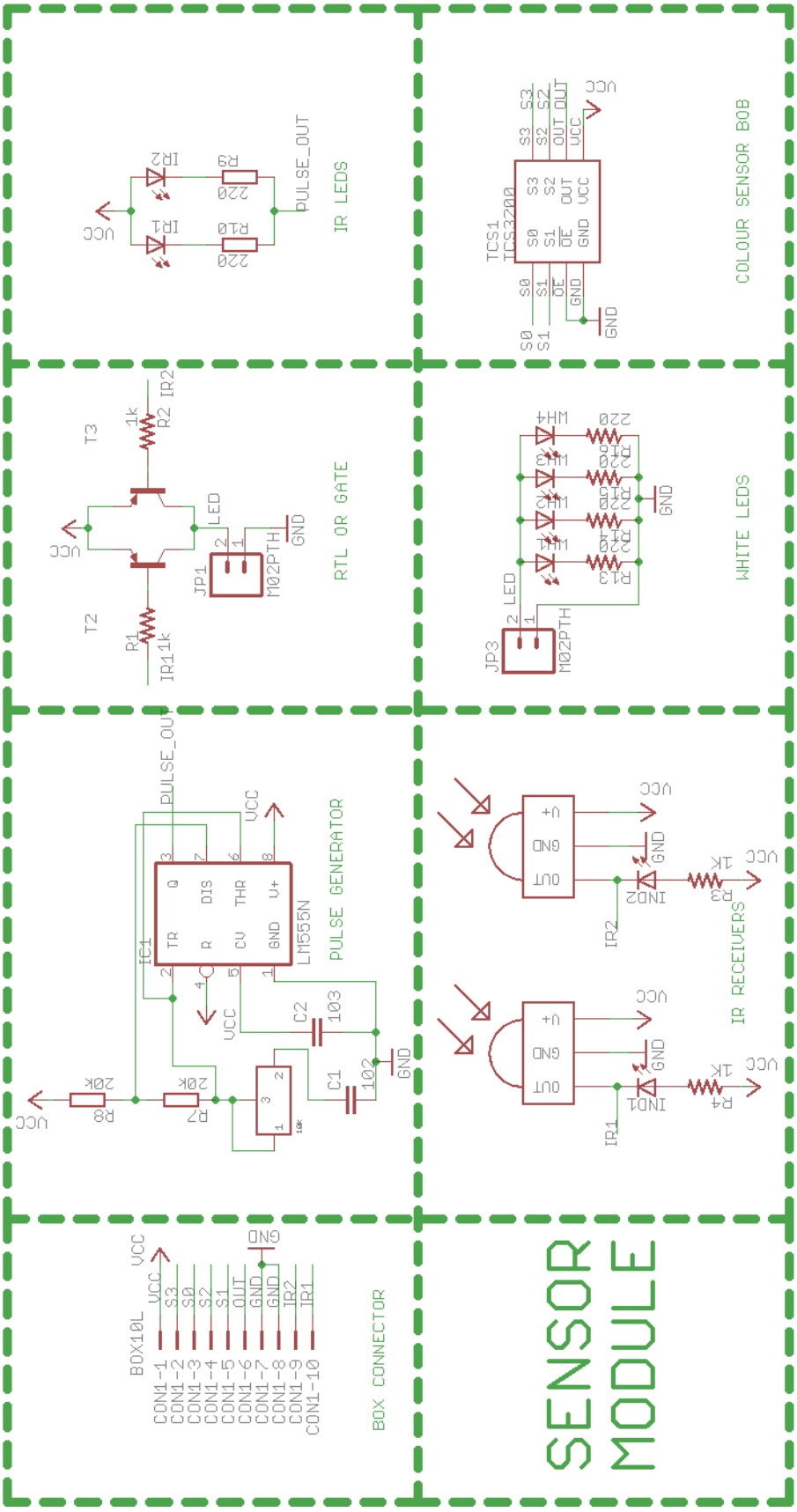
2.1.1 The Sensor Module

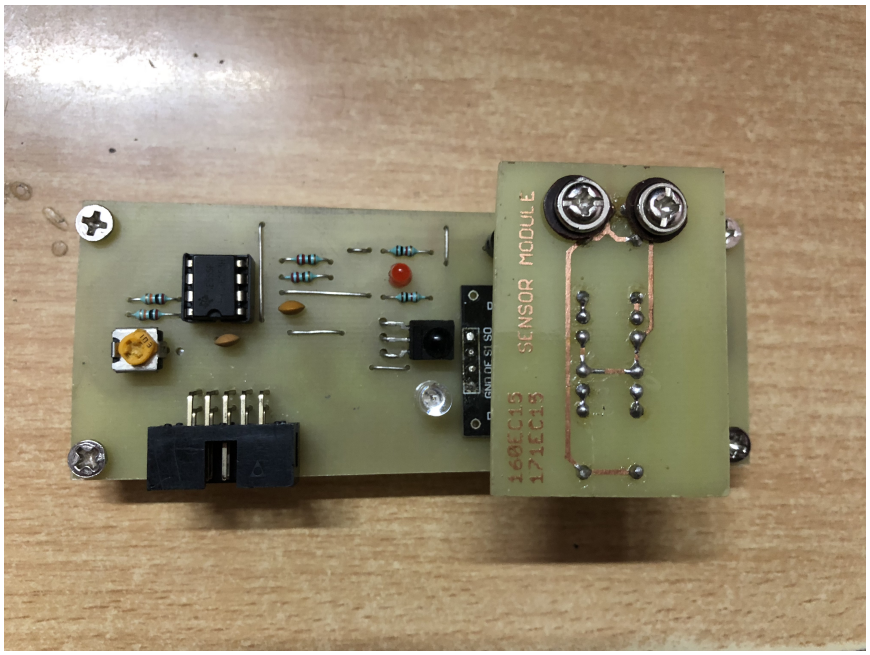
2.1.1.1 BOM

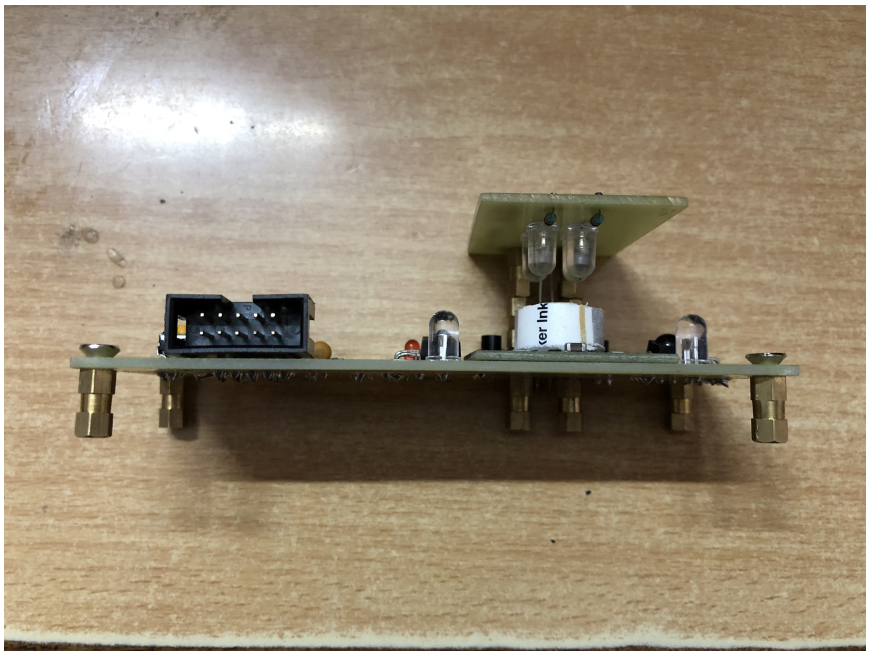
Part	Value	Device
C1	102	C-EU025-024X044
C2	103	C-EU025-024X044
CON1	BOX10L	BOX10L
IC1	LM555N	LM555N
IND1		LED3MM
IND2		LED3MM
IR1		LED5MM
IR2		LED5MM
JP1	M02PTH	M02PTH
JP3	M02PTH	M02PTH
R1	1k	RESISTOR0207/7
R2	1k	RESISTOR0207/7
R3	1K	RESISTOR0207/7
R4	1K	RESISTOR0207/7
R7	20k	R-EU0207/7
R8	20k	R-EU0207/7
R9	220	R-EU0207/7
R10	220	R-EU0207/7
R13	220	RESISTOR0207/7
R14	220	RESISTOR0207/7
R15	220	RESISTOR0207/7
R16	220	RESISTOR0207/7
T2		BC557
T3		BC557
TCS1	TCS3200BREAKOUT	TCS3200BREAKOUT
U1	TSOP348FLAT	TSOP348FLAT
U2	TSOP348FLAT	TSOP348FLAT
U3	10k	PRESETLR
WH1		LED5MM
WH2		LED5MM
WH3		LED5MM
WH4		LED5MM

Table 1: Bill of Materials for the Sensor Module

2.1.1.2 Schematic and the Board


SENSOR MODULE
BOX CONNECTOR
BOX10L
CON1-1
CON1-2
CON1-3
CON1-4
CON1-5
CON1-6
CON1-7
CON1-8
CON1-9
CON1-10
VCC
S3
S0
S2
S1
OUT
GND
GND
IR2
IR1
PULSE GENERATOR
IC1
LM555N
TR
Q
R
DIS
CV
THR
GND
V+
PULSE_OUT
C2
103
C1
102
R8
20K
R7
20K
IR RECEIVERS
IR2
IR1
IND2
IND1
R3
1K
R4
1K
OUT
GND
V+
WHITE LEDS
JP3
M02PTH
LED
R11
R12
R13
R14
220
GND
RTL OR GATE
T3
T2
R2
1k
R1
1k
JP1
M02PTH
LED
COLOUR SENSOR BOB
TCS1
TCS3200
S0
S1
S2
S3
OUT
OE
VCC
GND
IR LEDS
R9
R10
220
IR2
IR1
PULSE_OUT





Figures 2 and 3: Schematic of the Sensor Module,
Sensor Module Final Board After Soldering

2.1.1.3 Detailed device and component description The sensor module has 2 sensors on the board: TCS3200 colour sensor and two TSOP 34838 IR receivers. Apart from these, the sensor module has a 555 timer being operated in the astable mode, two IR LEDS and an assembly of 4 white LEDS. Four white LEDs together create a plane source as opposed to a point source created by using a single LED. The outputs of the two IR sensors are ORed using an RTL OR gate made from resistors and two PNP transistors(BC 557).

555 Timer: The 555 timer is connected in the astable configuration to provide a pulse output with frequency 38kHz. The presence of the 10k Ohm preset helps in adjusting the 555 frequency, which in turn determines the range for which the TSOP sensors will work as per our desire. The 555 output is connected to the input of the two IR LEDs, which makes them flash at this rate. The reason for this particular value is explained below.

TSOP 34838 IR Receivers: The block diagram of this sensor is given below. The PIN diode works as a photodetector and detects the light coming from the IR LEDs. The bandpass filter has a center frequency of 38kHz for 38348 and hence, the IR LEDs are made to flash at this rate. Whenever an obstruction is introduced, the pulsating light reflected from it falls on the sensor and makes its output go low.

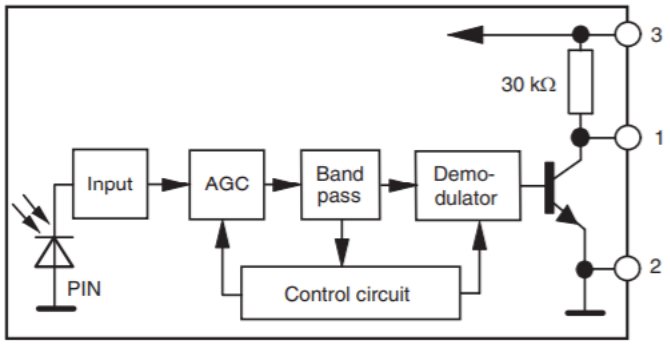


Figure 4: Block diagram of the TSOP sensor

Colour Sensor TCS3200: The pin diagram and block diagram of this sensor are given below.

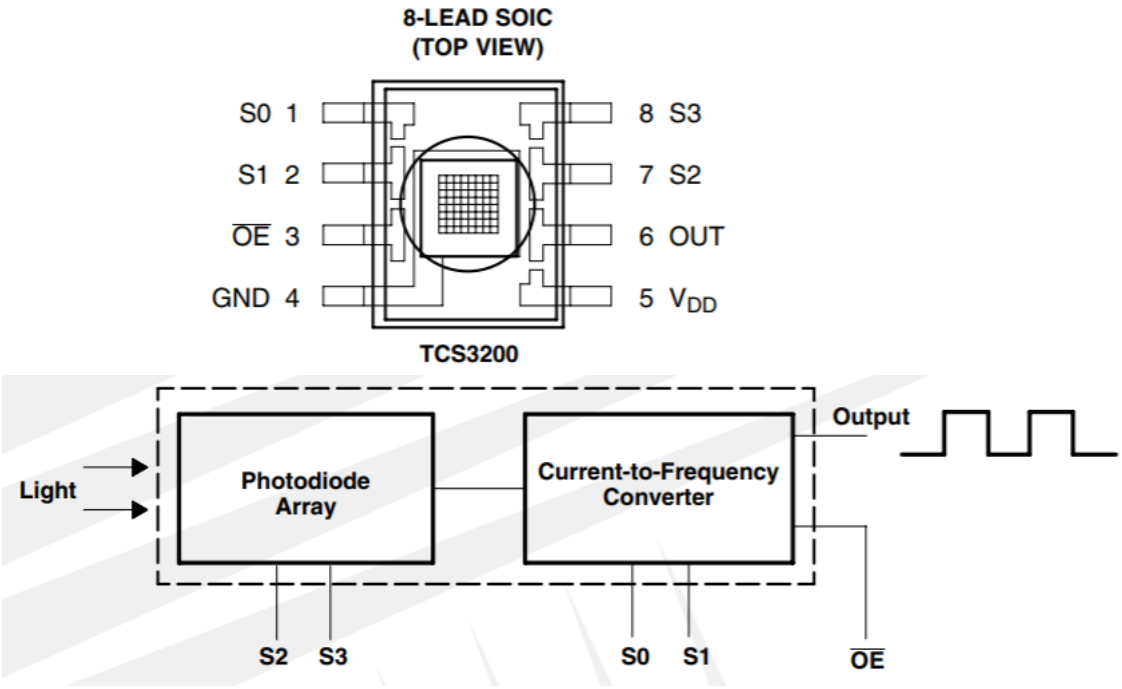


Figure 5: Pin and block diagrams of colour sensor

The TCS3200 is basically, a colour light to frequency converter. The 8X8 photodiode array detects the light falling on it and the output current of the photodiode array, which is proportional to the irradiance is converted into frequency by the current to frequency converter. Sixteen photodiodes have blue filters, 16 photodiodes have green filters, 16 photodiodes have red filters, and 16 photodiodes are clear with no filters. The output of this sensor is a square wave of 50 percent duty cycle whose frequency is proportional to the irradiance.
The pins S0 and S1 are used for output frequency scaling according to the table below.

S0	S1	Output Frequency Scaling(f0)
L	L	Power Down
L	H	2 %
H	L	20 %
H	H	100 %

Table 2: Function of pins S0 and S1

The pins S2 and S3 are set to get the output square wave for different primary colors as follows:

S2	S3	Photodiode Type
L	L	Red
L	H	Blue
H	L	Clear (No filter)
H	H	Green

Table 3: Function of pins S2 and S3

Circuit Description: Whenever a note is introduced in the space between the white LEDs and the sensors, the TSOP sensors detect an obstruction and the IR1 or IR2 line goes low. This triggers the backlight LEDs, as they are hardwired to turn on when either one of these signals goes low. The direction from which the note has entered can be inferred on the basis of which signal went low first. White light from the LED gets filtered by the note and falls on the colour sensor, which can now be read to infer its colour.

2.1.2 The Motherboard

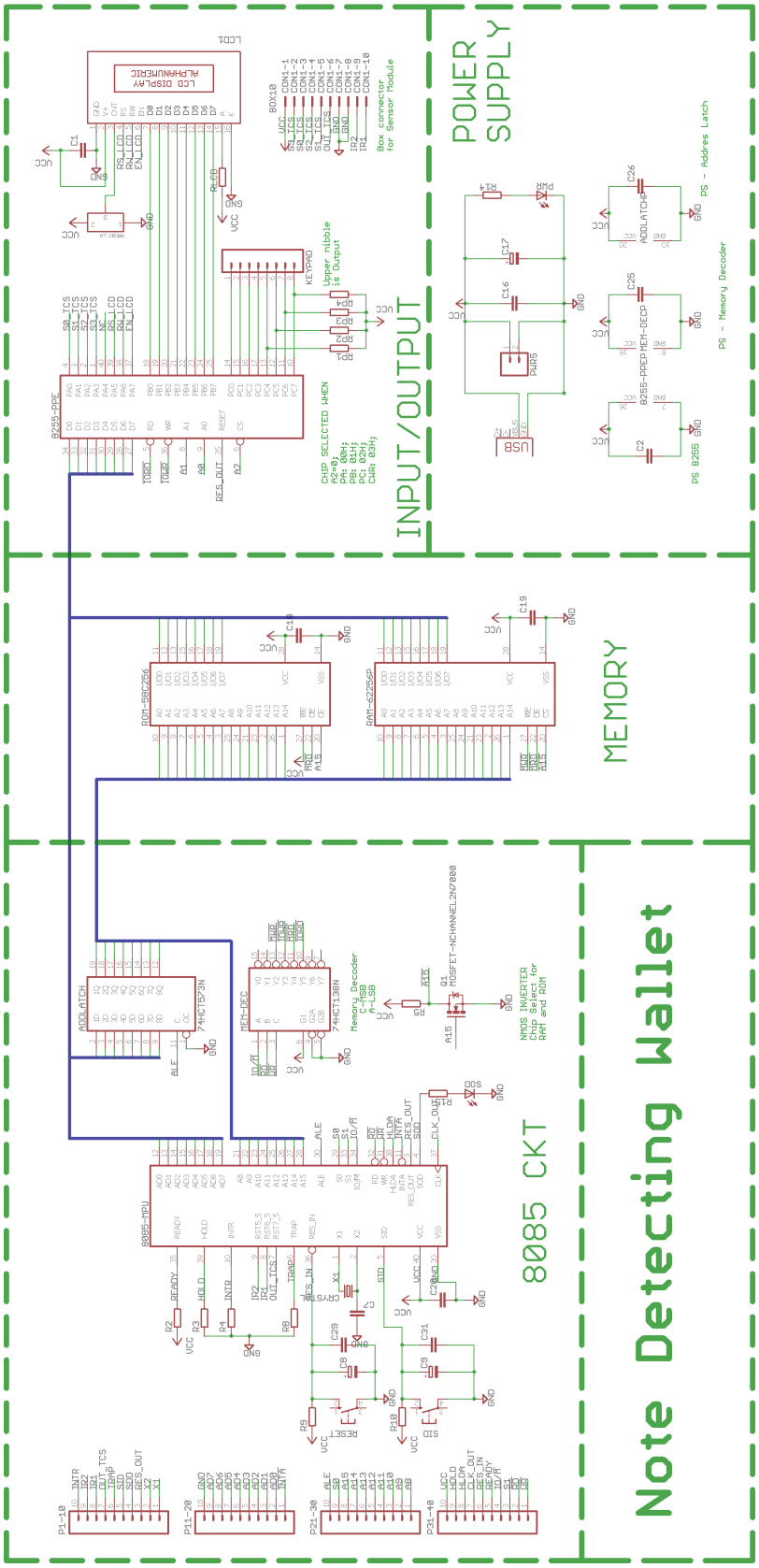
2.1.2.1 BOM

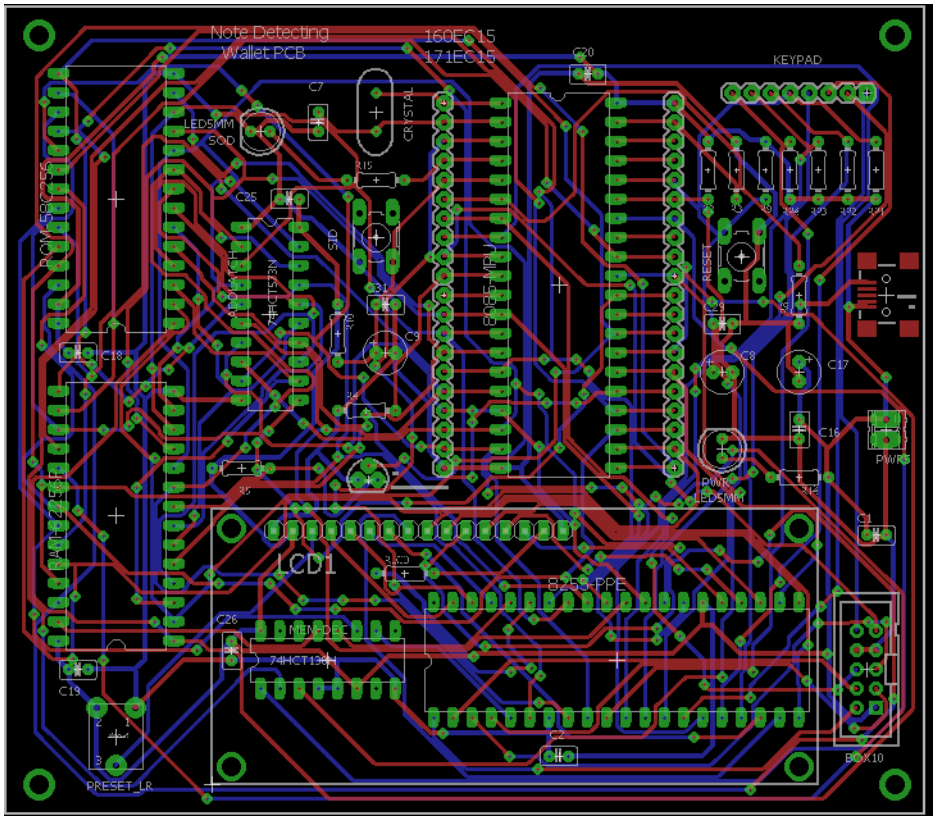
Part	Value	Device
8085-MPU		8085
8255-PPE		8255A
ADDLATCH	74HCT573N	74HCT573N
C1		C-EU025-025X050
C2		C-EU025-025X050
C7		C-EU025-025X050
C8		CPOL-EUE2.5-6
C9		CPOL-EUE2.5-6
C16		C-EU025-025X050
C17		CPOL-EUE2.5-6
C18		C-EU025-025X050
C19		C-EU025-025X050
C20		C-EU025-025X050
C25		C-EU025-025X050
C26		C-EU025-025X050
C29		C-EU025-025X050
C31		C-EU025-025X050
CON1	BOX10	BOX10
CRYSTAL		CRYSTALHC49US
KEYPAD		M08SILKFEMALEPTH
LCD1		16X2LCD
MEM-DEC	74HCT138N	74HCT138N
P1-10		M10SILKFEMALEPTH
P11-20		M10SILKFEMALEPTH

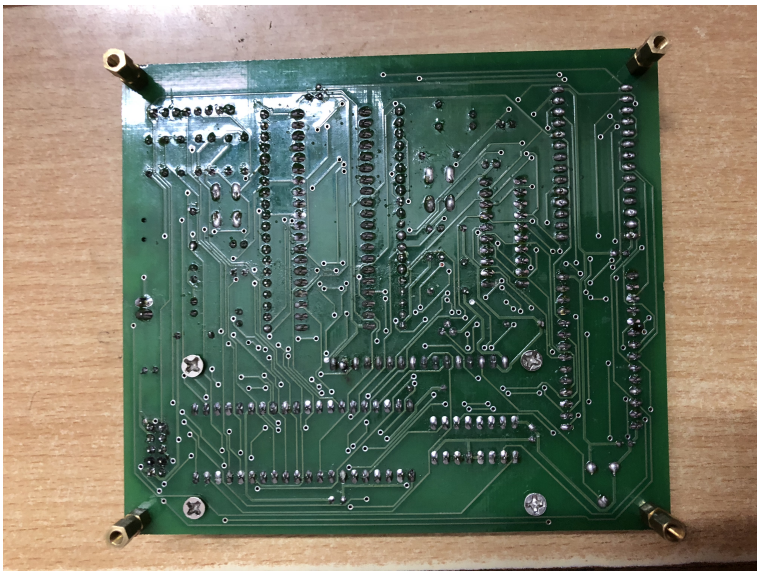
P21-30		M10SILKFEMALEPTH
P31-40		M10SILKFEMALEPTH
PWR	LED5MM	LED5MM
PWR5		M02
Q1	MOSFET-NCHANNEL2N7000	MOSFET-NCHANNEL2N7000
R2		R-EU0204/7
R3		R-EU0204/7
R4		R-EU0204/7
R5		R-EU0204/7
R8		R-EU0204/7
R9		R-EU0204/7
R10		R-EU0204/7
R14		R-EU0204/7
R15		R-EU0204/7
RAM-62256P		62256P
RESET		10-XX
RLCD		R-EU0204/7
ROM-58C256		58C256P
RP1		R-EU0204/7
RP2		R-EU0204/7
RP3		R-EU0204/7
RP4		R-EU0204/7
SID		10-XX
SOD	LED5MM	LED5MM
U1	PRESETLR	PRESETLR
X1	USBSMD	USBSMD

Table 4: Bill of materials for the motherboard

2.1.2.2 Schematic, Board Layout and the Board


Note Detecting Wallet
8085 CKT
MEMORY
INPUT/OUTPUT
POWER SUPPLY
LCD DISPLAY
AL-PHANUMERIC
KEYPAD
8255-PPI
8085-MPU
ADD LATCH
MEM-DEC
ROM-58C256
RAM-62256
PS - Address Latch
PS - Memory Decoder
PS 8255


Note Detecting
Wallet PCB
160EC15
171EC15
KEYPAD
CRYSTAL
LEDSMM
SOD
LCD1
8255-PPE
PRESET_LR
BOX10
PWR
LEDSMM



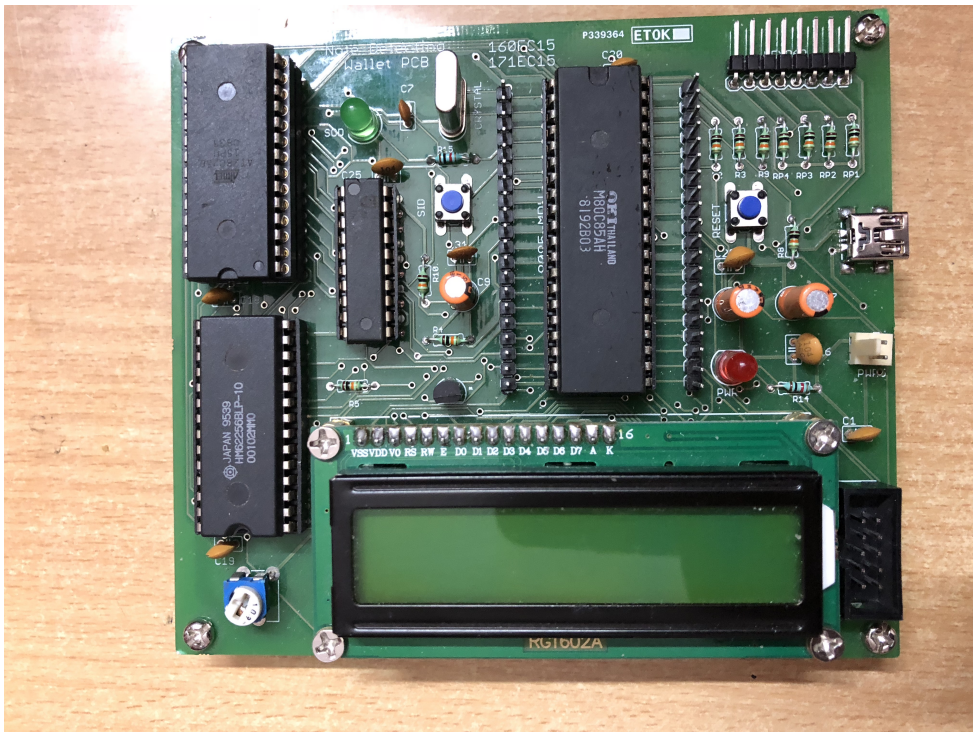


Figure 6: Final Soldered Motherboard

2.1.2.3 Detailed device and component description The motherboard can be broadly divided into four sections: 8085 microprocessor interfacing and timing, memory interfacing, I/O interfacing and the power supply.
8085 μP: The schematic diagram for this section is shown below.

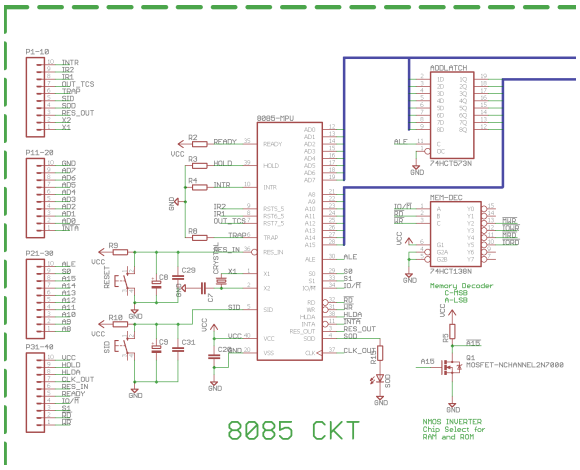


Figure 7: Microprocessor interfacing section

1. The Clock
The maximum allowed frequency for operation in 8085AH is 5 MHz. A source crystal of 10MHz has been used. Therefore, the system will work at 5 MHz clock frequency. Moreover, the 22pF capacitor used with crystal provides enough loading to crystal to generate oscillations.

2. Reset Switch
This switch will take back the whole system to its initial coordinates i.e. PC=0000H, whenever the system gets stuck while executing a code.

3. SID Switch and the SOD LED
These are the 1-bit serial communication pins of 8085. In our project, they are simply used to show communication between a push button and a red LED.

4. Interrupts
Out of the five interrupts which 8085 offers, we have used three in our project. RST7.5 is connected to the output of the colour sensor, and RST6.5 and RST5.5 are used to receive interrupt signals from the two TSOP sensors.

5. Demultiplexing of lower order address and data bus
In 8085, the lower order address and data bus i.e. AD7-AD0 are multiplexed. In order to demultiplex these signals, we have used 74573 which is an octal D-latch with non-inverting outputs. The ALE signal is used to enable the latch. ALE goes high at the beginning of every machine cycle, when AD7-AD0 carries the lower order address. In this phase, the latch is enabled and hence, the lower order address gets latched at the outputs of 573. When ALE goes low, the data becomes available at AD7-AD0.

6. Generation of control signals
The generation of read and write signals for I/O and memory has been done with the help of the 3X8 decoder, 74138. Inputs to the decoder are RD*, WR* and IO/M*, and after decoding of all these signals, we have four useful signals which would help us in enabling the I/O and memory peripherals as and when required viz. MR* (Memory Read), MW* (Memory Write), IOR* (I/O Read) and IOW* (I/O Write).

Memory Interfacing: The schematic diagram for this section is shown below.

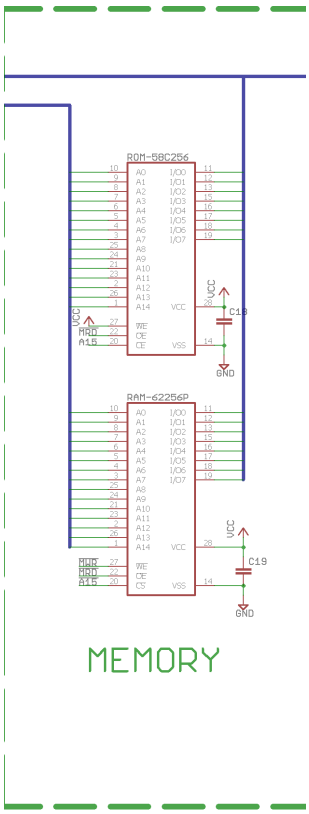


Figure 7: Memory interfacing section

RAM is a volatile memory which loses its data when power is turned off. Since we want the system code to remain saved, therefore a non-volatile memory ROM is also needed. We have used a 32K RAM 62256P and 32K ROM 58C256 in this project. The decision to use these particular memories has been influenced by the requirement of keeping the hardware minimal. The way that this scheme is successful in achieving this can be understood if we analyze the address decoding for the two memories. 8085 offers 16 address lines and hence, a total of 2^{16} memory locations can be addressed i.e. 64K bytes of memory. Hence, we use 32K of ROM and 32K of RAM, each of which requires 15 address lines for memory locations and 1 address line i.e. A15 can be used for chip selection.

After every power off or reset, the program counter directs to the location 0000H. So, we need the program to be stored at 0000H and hence, we select the ROM starting address as 0000H. When A15 is 0, ROM is selected and when A15 is 1, RAM is selected. Now comes the point of hardware reduction. Instead of using an entire NOT gate IC which contains 6 NOT gates, we have used an N-channel

MOSFET BS170 to obain A15 and A15*.

I/O Interfacing: The schematic diagram for this section is shown below.

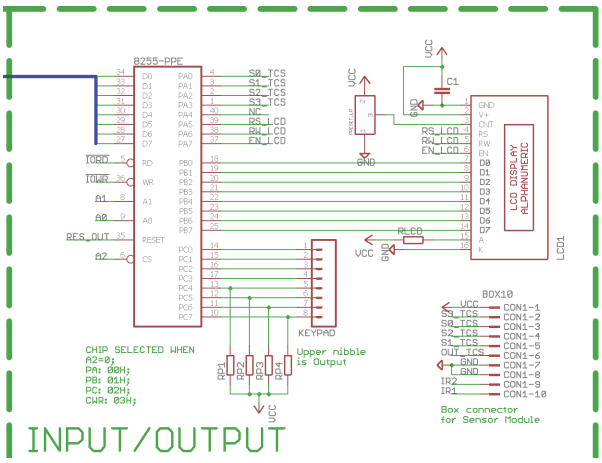


Figure 8: I/O interfacing section

As per the system block diagram given earlier, we have connected 3 input devices and 1 output device with 8085. Since 2 input devices i.e. TSOP sensors and the colour sensor have been integrated on the sensor module, so we have considered it as one input device in the further report. Therefore,

Input Devices: Sensor Module and 4X4 Switch Keypad
Output Device: LCD

The sensor module has been connected to the motherboard using a simple 10-pin box connector. In order to interface the LCD and keypad and a few signals from the box connector, we have used a programmable peripheral interface, Intel 8255A.

The 8255A PPI: 8255A is a 40-pin DIP programmable peripheral interface IC specially designed to communicate with Intel μP. It has 3 I/O ports named PORT A, PORT B and PORT C. We have used these ports as follows:

PORT A: We call this the 'control' port, as it manages all the I/O control. It has 7 signals connected to its pins; the three LCD control signals and four colour sensor control signals (S0, S1, S2 and S3). It functions as an output port.
PORT B: This is connected to data pins of the LCD. It also functions as an output port.
PORT C (upper): Keypad Output.
PORT C (lower): Keypad Input.

The 8255A can work in 3 modes. The first is MODE 0, which is simple I/O with

no handshaking. The second mode is MODE 1, in which PORT A and PORT B use handshake signals from PORT C to communicate with the peripherals. The third and last mode is MODE 2, in which PORT A is bidirectional, PORT B can be in MODE 0/1 and PORT C is used for handshaking.

In our case, 8255A works in MODE 0. Chip selection of 8255A is done through pin A2. When A2=0, then 8255 will get selected. The 8255 data pins are connected to the 8085 data bus. I/O read and write control signals are provided by the decoded control signals explained in last section i.e. IOR* and IOW*. The Reset pin of 8255A is connected to Reset Out Pin of 8085, so that whenever 8085 is reset, then all the associated peripherals also get reset.

Power Supply: The schematic diagram for this section is shown below.

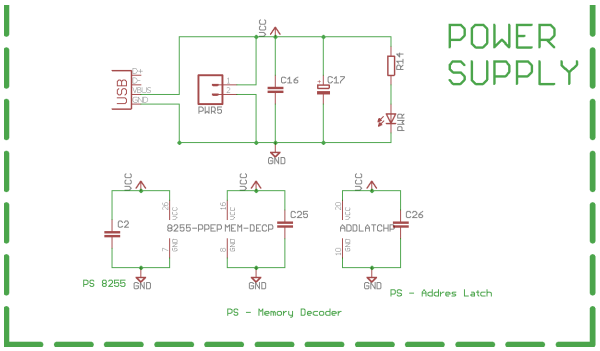


Figure 9: Power supply section

8085 and other associated peripherals use +5V power supply. This has been provided using a mini USB, which can be connected to a laptop or a power bank to draw power. Power on is indicated by the power LED. The two capacitors C16 and C17 are used for low and high frequency noise filtering and R14 is used as current limiting resistor for the LED.

2.1.3 Designing and Fabrication of boards

All the schematics and board layouts were made using Eagle CAD. The sensor module PCB was fabricated in the CEDT lab. The gerber files for the motherboard were sent to PCB Power House for fabrication.

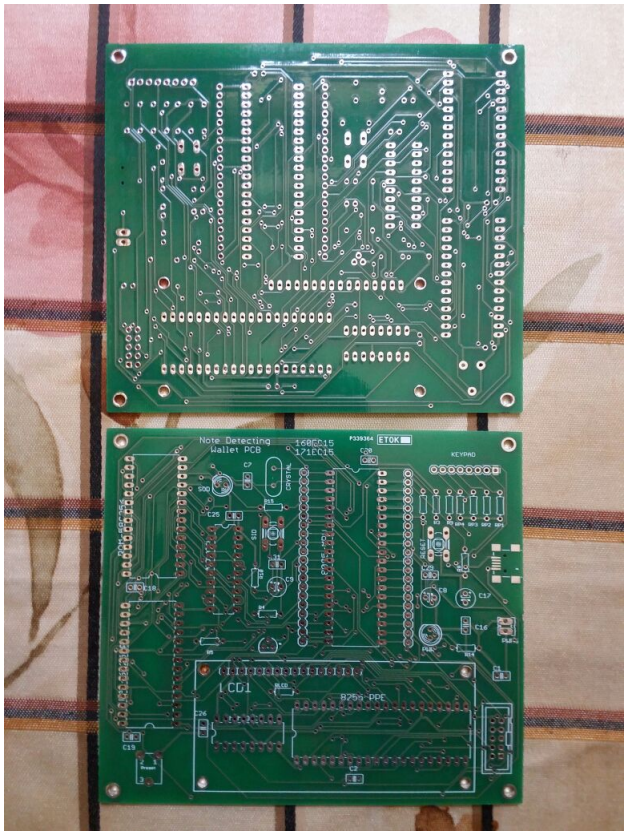


Figure 10: Motherboard after coming from PCB Power House

2.2 Software

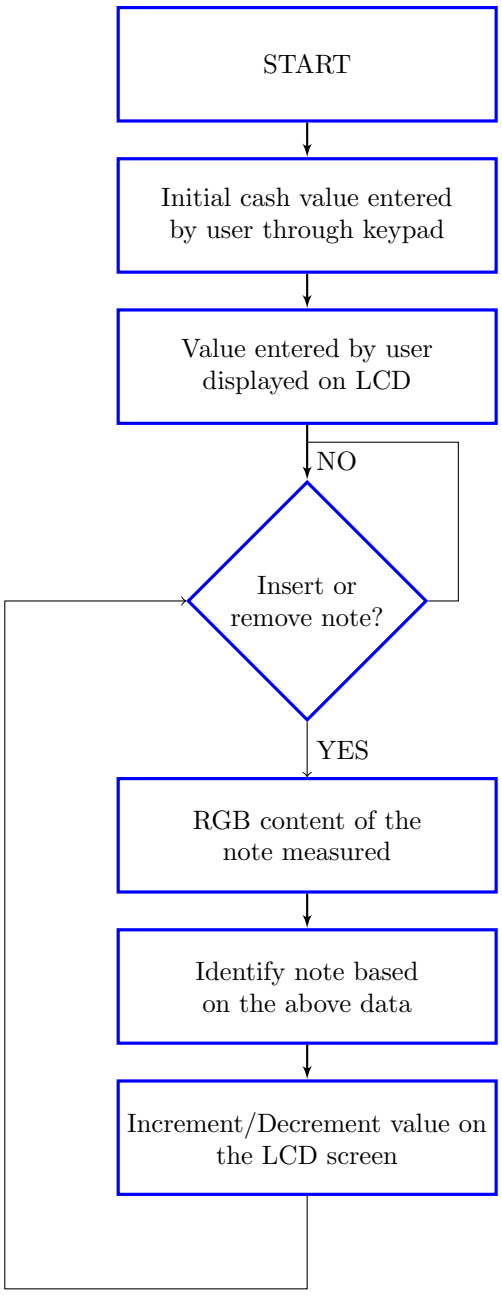


Figure 11: Flow Chart of the Software

2.2.1 Assembly Code of the main algorithm

```
DETECT_NOTE:            PUSH PSW
                        PUSH D
                        PUSH H                  ;STORE RAW DATA POINTER ON
                                                STACK
                        LXI H, 8011H            ;NOTE COUNT LOOP
                        MVI M, 08H
                        LXI H, NOTE_DATABASE
                        INX H                   ;1 INDEXED ARRAY
                        SHLD 8015H              ;INITIALISE POINTER TO TABLE
DETECT_NOTE
_MAIN_LOOP:             MVI A,FFH               ;INITIALISE BOOLEAN
                        LXI H, 8010H            ;COUNT COLOUR LOOP
                        MVI M, 03H
DETECT_NOTE
_INNER_LOOP:            POP H                   ;LOAD RAW DATA POINTER
                        MOV C,M                 ;BC = RAW COLOUR
                        INX H
                        MOV B,M
                        INX H
                        PUSH H                  ;STORE RAW DATA ADDRESS ON
                                                STACK
                        PUSH B                  ;STORE RAW DATA ON STACK
                        LHLD 8015H              ;LOAD POINTER TO TABLE
                        MOV E,M                 ;DE = COLOUR_THRESHOLD_LOW
                        INX H
                        MOV D,M
                        INX H
                        CALL BC_GREATER_THAN_DE ;RETURNS IN B
                        ANA B
                        MOV E,M                 ;DE = COLOUR_THRESHOLD_HIGH
                        INX H
                        MOV D,M
                        INX H
                        SHLD 8015H              ;STORE POINTER
                        POP B                   ;LOAD RAW DATA FROM STACK
                        CALL BC_LESS_THAN_DE    ;RETURNS IN B
                        ANA B
                        LXI H, 8010H            ;COUNT
                        DCR M
                        JNZ DETECT_NOTE_INNER_LOOP ;EXIT IN 3 ITERATIONS
                        ANA A                   ;READ BOOLEAN
```

```
                        JNZ DETECT_NOTE_NOTE_FOUND   ;BRANCH IF TRUE
DETECT_NOTE
_INNER_LOOP_END:        POP H                        ;POINTS TO INVALID ADDRESS
                                                     PAST RAW DATA
                                                     ;SHIFT POINTER TO VALID RAW
                                                     DATA AGAIN
                        LXI D,FFFAH                  ;2'S COMPLEMENT OF 6
                        DAD D                        ; DECREMENT HL BY 6
                        PUSH H
                        LHLD 8015H                   ;SKIP NOTE VALUE IN LOOKUP
                                                     TABLE
                        INX H
                        INX H
                        SHLD 8015H                   ;POINTS TO RED_LOW_LOW NOW
                        LXI H,8011H                  ;NOTE COUNT
                        DCR M
                        JNZ DETECT_NOTE_MAIN_LOOP    ;EXIT IN 8 ITERATIONS
DETECT_NOTE
_NOTE_NOT_FOUND:        LXI B,0000H                  ;RETURN 0
                        JMP DETECT_NOTE_END
DETECT_NOTE
_NOTE_FOUND:            LHLD 8015H                   ;LOAD POINTER TO TABLE
                        MOV C,M                      ;BC = VALUE
                        INX H
                        MOV B,M
DETECT_NOTE
_END:                   POP H                        ;LOAD BACK RAW DATA
                        POP D
                        POP PSW
                        RET
```

2.2.2 Explanation and pseudo-code for the assembly code

We identify a colour by comparing it with two threshold values. If the value of the raw colour data obtained from the sensor is greater than the low threshold, but less than the high threshold, the raw data is classified as the given colour. Our database is an array of note structures each of which stores the high and low threshold values of red, green and blue colours, and the value of the note. This database has been filled manually with a lot of experimentation.
The raw RGB colour values from the sensor are the input to the detectNote function. It then compares these to those of each note stored in the data-base linearly, and returns the value of the note when a match is found.

```
struct note{
     int redLow, redHigh;
     int greenLow, greenHigh;
     int blueLow, blueHigh;
     int value;
}
int detectNote(int red, int green, int blue){
     for(int i = 0; i <= noteArray.length(); i++){
         struct note referenceNote = noteArray[i];
         bool A = true;
         A = A && (red > referenceNote.redLow);
         A = A && (red < referenceNote.redHigh);
         A = A && (red > referenceNote.greenLow);
         A = A && (red < referenceNote.greenHigh);
         A = A && (red > referenceNote.blueLow);
         A = A && (red < referenceNote.blueHigh);
         if(A == true)
             return referenceNote.value
}
     return 0;
}
```

3 Results

3.1 Note-wise final output

We took seven denominations of Indian currency notes viz. 10, 20, 50 (new and old), 100, 200, 500 and 2000 and conducted 10 trials for each note using the project set up. The percentage of correct detection has been given in the table below.

Note	Percentage of Correct Detection
10	100%
20	90%
50 (old)	50%
50 (new)	70%
100	70%
200	100%
500	50%
2000	70%

3.2 Final Working Project

Figure 12: 100 Rupees testing

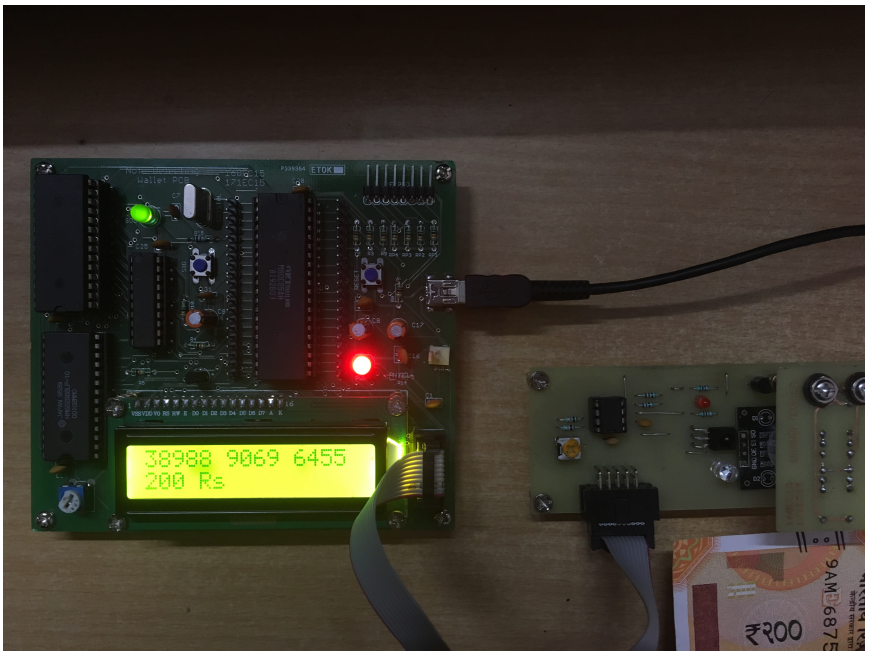


Figure 13: 200 Rupees testing

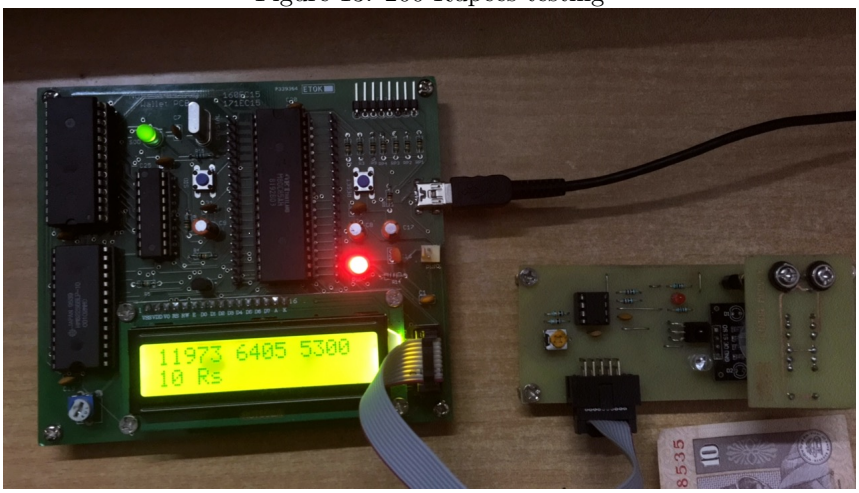


Figure 14: 10 Rupees testing

4 Limitations of the project

1. The project does not work efficiently for used notes as the data given by the colour sensor changes drastically.

2. The project does not work for all sides and all points on the note, because of the same reason as above.

3. Ambient light can affect the readings easily.

4. Notes with similar colour signatures were difficult to detect because of close readings.

These problems can possibly be overcome by using a larger data set and a smarter algorithm. Since this is a pattern matching problem, it is within the domain of a neural network.

References

[1] Gaonkar,R.S. *Microprocessor Architecture, Programming, and Applications with the 8085.* Prentice Hall, 2002. isbn: 9780130195708.

[2] http://8085projects.in/

[3] *Sorting Machine.* URL: https://github.com/MCodez/Sorting-Machine-Intel-8085Microprocessor-Project

[4] Mohamed, Aisah and Ikram Ishak, Mohd and Buniyamin, Norlida, 2012, *Development of a Malaysian Currency Note Recognizer for the Vision Impaired*, isbn: 978-1-4577-1965-3, 2012 Spring World Congress on Engineering and Technology, SCET 2012 - Proceedings.

[5] Sahu, Snehlata and Verma, Toran. *Identification of Paper Currency Techniques: A Survey*, IJSTE - International Journal of Science Technology and Engineering, Volume 2, Issue 12, June 2016.