Candy Vending Machine.

Using 8051 Microcontroller

J Component

ECE 3031-MicroControllers and Embedded System

Submitted By

17BEC0179	**V.YESWANTH**
17BEC0172	**K.B.S.MARUTHI RAJU**
17BEC0557	**GUNA AVINASH**

To

Prof Karthikeyan B Sir

School of Electronics Engineering

VIT University,Vellore

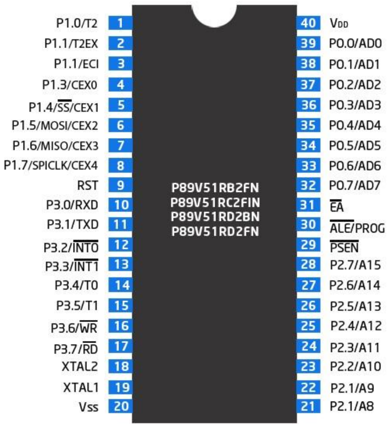
Tamil Nadu - 632014

Abstract:

The project deals with the real time application of the 8051 microcontroller as a candy dispenser. Involves interfacing LCD, motor and indirect interfacing of RFID reader with Arduino UNO which in turn parallelly communicates with 8051. Basically, the Arduino Uno is used as slave and 8051 as the master. The 8051 is programmed entirely in assembly language to control LCD,running time of motor and RFID. All the instructions are displayed on the LCD controlled by the microcontroller. Three types of cards, two cards with different amounts and an invalid card are chosen for demonstrating the working model. The RFID connected to Arduino reads the cards and parallelly communicates to 8051 to display respective message. The customer can set the number of candies he/she wants with buttons provided to increment or decrement he value and corresponding value is displayed on the LCD by the 8051.

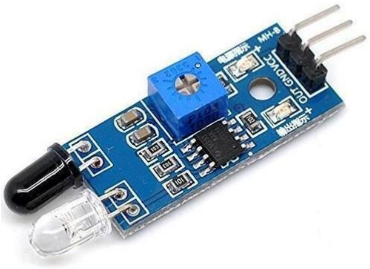
Components Required:

- **8051 Microcontroller(P89V51RD2):**



The NXP (founded by Philips) P89V51RD2 is a 40MHz, 5 Volt 8051-based Microcontroller with 32 I/O lines, 3 Timers/Counters, 9 Interrupts/4 priority levels, 64K+8K FLASH, 1K on-chip RAM, SPI, Dual Data Pointers, WDT, 5-channel PCA.

- **IR SENSOR**



IR Sensors work by using a specific light **sensor** to detect a select light wavelength in the Infra-Red (**IR**) spectrum. By using an LED which produces light at the same wavelength as what the **sensor** is looking for, you can look at the intensity of the received light.

- **UNO:**

Arduino is an open-source electronics platform based on easy-to-use hardware and software. **Arduino** boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online.

- **Servo-Motor**

A **servo motor** is an electrical device which can push or rotate an object with great precision. If you want to rotate and object at some specific angles or distance, then you use **servo motor**. It is just made up of simple **motor** which run through **servo** mechanism

(*)**KEYPAD**

Introduction:

The microcontrollers used in the project are Aryabhata 8051 AT89S52 and Arduino UNO. The software for programming code for 8051 is Keil uVision5 and programmer ISP for burning the program HEX file into the microcontroller. The Arduino IDE v1.8.9 is used for programming Arduino Uno board. A relay module with control pin connected to 8051 to control the running time of the motor and other necessary components like 220 uF capacitor and voltage regulator used. The setup is made such that, the necessary messages are displayed in LCD like “Place RFID” or “Enter the amount” and the customer’s RFID is read after placing it on the RFID reader. Now, connections are made so that Arduino Uno and the 8051 can communicate parallelly with each other where Arduino sets particular bits and 8051 checks for those bits for if they are set or not to act accordingly and vice versa whenever the RFID reader detects the card placed. The keypad are used to set the number of Candies and the number is displayed in the LCD by the 8051 and then timer is programmed in 8051 to decide the running time of the motor to control the number of Candies dispensed.

Code Written in Keil IDE for 8051:

```
ORG 00H
    Q:CLR P3.6
    CLR P3.7
    CLR P3.4
    MOV A,#38H
    ACALL CMD                              ;To send command to LCD
    MOV A,#0FH                             ;To display on cursor blinking
    ACALL CMD
    MOV A,#01H                             ;clear display
    ACALL CMD
    MOV A,#06H                             ;Increment cursor
    ACALL CMD
    MOV A,#80H                             ;force cursor to 1st line
    ACALL CMD
    MOV DPTR,#400H
    MOV R4,#10
    LOOP:CLR A
    MOVC A,@A+DPTR
    ACALL D
    INC DPTR
    DJNZ R4,LOOP
    ROP:JB P3.6,JUMP              ;To check Rs5 or Rs.10 card placed
    JB P3.7,JUMP1
```

```
SJMP ROP
JUMP1:MOV A,#01H
ACALL CMD
MOV A,#06H
ACALL CMD
MOV A,#80H
ACALL CMD
MOV DPTR,#650H
MOV R4,#12
LOO:CLR  A
MOVC A,@A+DPTR
ACALL D
INC DPTR
DJNZ R4,LOO
LJMP Q
JUMP:MOV A,#01H
ACALL CMD
MOV A,#06H
ACALL CMD
MOV A,#80H
ACALL CMD
MOV DPTR,#600H
MOV R4,#16
LO:CLR A
MOVC A,@A+DPTR
ACALL D
INC DPTR
DJNZ R4,LO
```

```
    MOV A,#01H
    ACALL CMD
    MOV A,#06H
    ACALL CMD
    MOV A,#80H
    ACALL CMD
    CLR A
BACK:MOV P1,#11111111B // loads P1 with all 1's
     CLR P1.0  // makes row 1 low
     JB P1.4,NEXT1  // checks whether column 1 is low and
jumps to NEXT1 if not low
     MOV A,#1D   // loads a with 0D if column is low (that
means key 1 is pressed)
     ACALL SER
     ACALL DISPLAY  // calls DISPLAY subroutine
NEXT1:JB P1.5,NEXT2 // checks whether column 2 is low and
so on...
      MOV A,#2D
      ACALL SER
      ACALL DISPLAY
NEXT2:JB P1.6,NEXT3
      MOV A,#3D
      ACALL SER
      ACALL DISPLAY
NEXT3:JB P1.7,NEXT4
         LJMP T
NEXT4:SETB P1.0
      CLR P1.1
      JB P1.4,NEXT5
      MOV A,#4D
      ACALL SER
      ACALL DISPLAY
NEXT5:JB P1.5,NEXT6
      MOV A,#5D
      ACALL SER
      ACALL DISPLAY
NEXT6:JB P1.6,NEXT7
      MOV A,#6D
      ACALL SER
      ACALL DISPLAY
NEXT7:JB P1.7,NEXT8
NEXT8:SETB P1.1
      CLR P1.2
      JB P1.4,NEXT9
      MOV A,#7D
      ACALL SER
      ACALL DISPLAY
NEXT9:JB P1.5,NEXT10
      MOV A,#8D
      ACALL SER
      ACALL DISPLAY
NEXT10:JB P1.6,NEXT11
       MOV A,#9D
       ACALL SER
       ACALL DISPLAY
NEXT11:JB P1.7,NEXT12
NEXT12:SETB P1.2
```

```
       CLR P1.3
       JB P1.4,NEXT13
NEXT13:JB P1.5,NEXT14
         MOV A,#0D
         ACALL SER
      ACALL DISPLAY
NEXT14:JB P1.6,NEXT15
NEXT15:JB P1.7,BACK
       LJMP BACK
    T:MOV A,#55D
    ACALL SER
    MOV DPTR,#800H
         MOV R4,#12
      MOV A,#01H
    ACALL CMD
    MOV A,#06H
    ACALL CMD
    MOV A,#80H
    ACALL CMD
    L:CLR A
         MOVC A,@A+DPTR
         ACALL D
         INC DPTR
         DJNZ R4,L
         OP:JNB P3.4,OP
         LJMP Q
         K: SJMP K
    SER:MOV TMOD,#20H
```

```
 MOV TH1,#-3
 MOV SCON,#50H
 SETB TR1
 SE:MOV SBUF,A
 W:JNB TI,W
 CLR TI
 RET
CMD:MOV P2,A
CLR  P3.5
CLR P3.2
SETB P3.3
CLR P3.3
ACALL DELAY
RET
DISPLAY:MOV R7,#30H
ADD A,R7
MOV P2,A           ;Display command for LCD
SETB P3.5
CLR P3.2
SETB P3.3
CLR P3.3
ACALL DELAY
RET
D:MOV P2,A            ;Display command for LCD
SETB P3.5
CLR P3.2
SETB P3.3
CLR P3.3
```

```
    ACALL DELAY
    RET
    DELAY:MOV R3,#0FFH        ;Delay function in LCD
printing
    GO:MOV R2,#0FFH
    GO1:DJNZ R2,GO1
    DJNZ R3,GO
    RET
    here:sjmp here
    ORG 400H
        DB "PLACE RFID"
    ORG 600H
    DB "ENTER NO OF CHOC"
        ORG 650H
            DB "INVALID CARD"
    ORG 800H
    DB "COLLECT CHOC"
     END
```

Code written in Arduino IDE for Arduino UNO:

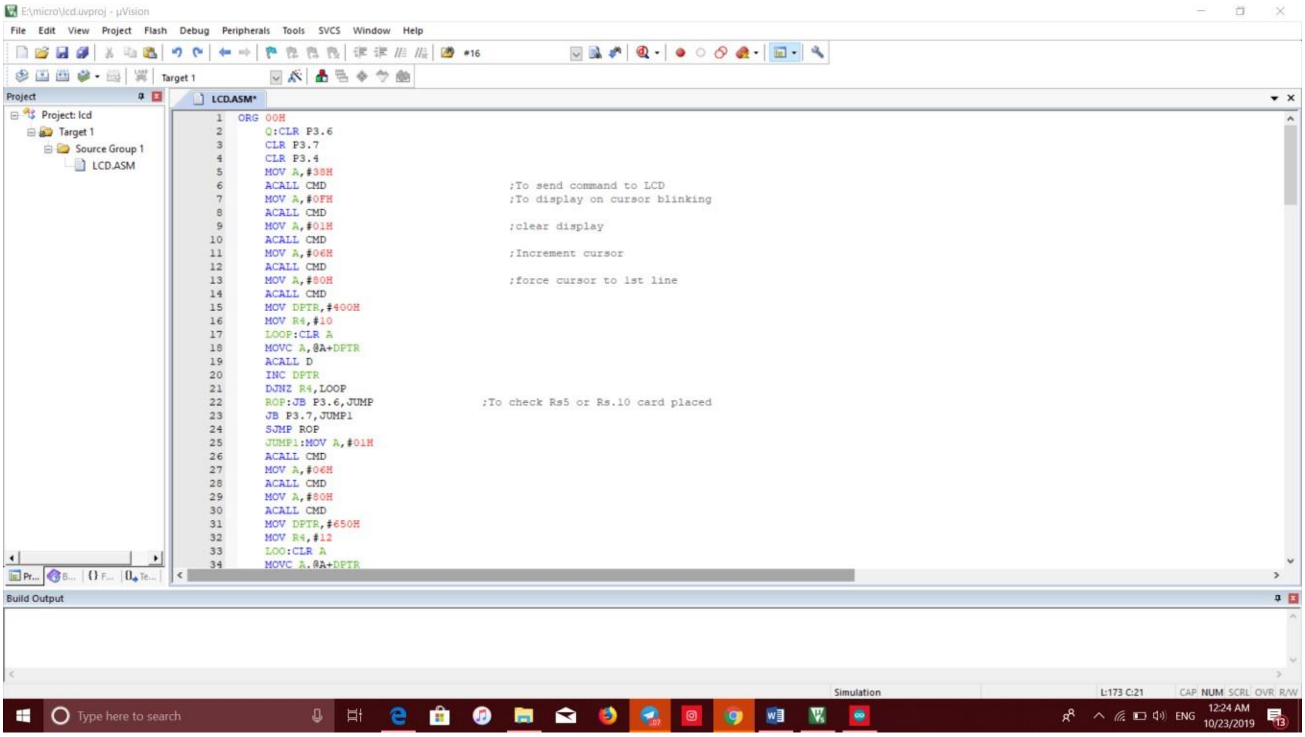
```
#include "SPI.h"
#include "MFRC522.h"
#include <Servo.h>
// Declare the Servo pin
int servoPin = 8;
// Create a servo object
Servo Servo1;
#define SS_PIN 10
#define RST_PIN 9
```

```
#define SP_PIN 8
int ProxSensor=2;
int inputVal=0;
MFRC522 rfid(SS_PIN, RST_PIN);

MFRC522::MIFARE_Key key;
int incomingByte = 0;
String str = "";
int a=0;
String in="A8:0F:9F:59";
void setup() {
  pinMode(2,INPUT);
  pinMode(7,OUTPUT);
  pinMode(6,INPUT);

  pinMode(3,OUTPUT);
  pinMode(4,OUTPUT);
  Serial.begin(9600);
  SPI.begin();
  rfid.PCD_Init();
  pinMode(ProxSensor,INPUT);
  Servo1.attach(servoPin);
  Servo1.write(0);
}
void loop() {
   digitalWrite(2,LOW);
  digitalWrite(7,LOW);
 digitalWrite(3,LOW);
```

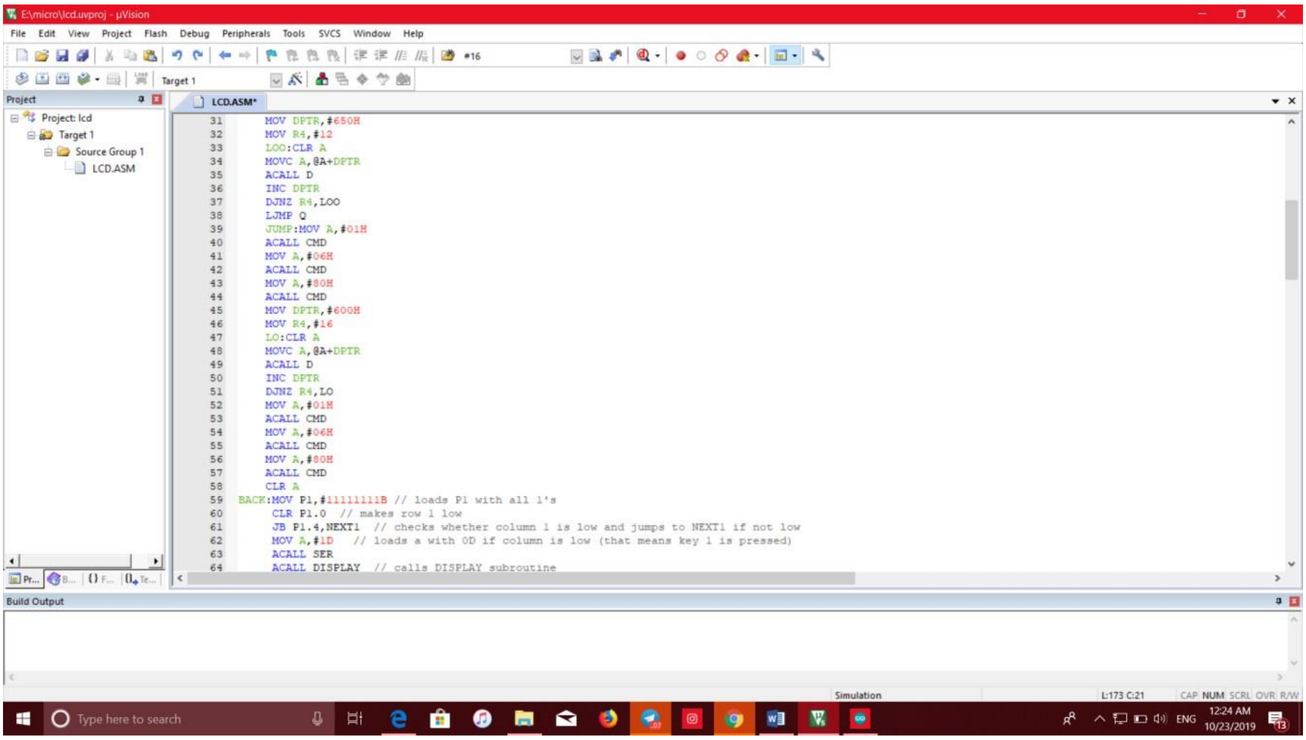
```
  digitalWrite(4,LOW);
  if (!rfid.PICC_IsNewCardPresent() ||
!rfid.PICC_ReadCardSerial())
    return;

  // Serial.print(F("PICC type: "));
  MFRC522::PICC_Type piccType =
rfid.PICC_GetType(rfid.uid.sak);
  // Serial.println(rfid.PICC_GetTypeName(piccType));

  // Check is the PICC of Classic MIFARE type
  if (piccType != MFRC522::PICC_TYPE_MIFARE_MINI &&
    piccType != MFRC522::PICC_TYPE_MIFARE_1K &&
    piccType != MFRC522::PICC_TYPE_MIFARE_4K) {
    Serial.println(F("Your tag is not of type MIFARE
Classic."));
    return;
  }

  String strID = "";
  for (byte i = 0; i < 4; i++) {
    strID +=
    (rfid.uid.uidByte[i] < 0x10 ? "0" : "") +
    String(rfid.uid.uidByte[i], HEX) +
    (i!=3 ? ":" : "");
  }
  strID.toUpperCase();

 // Serial.print("Tap card key: ");
```

```
   Serial.println(strID);
     if(strID==in){
    Serial.println("okay1");
    digitalWrite(7,HIGH);
   delay(2000);
   digitalWrite(7,LOW);
   while(1)
   {
  if (Serial.available() > 0) {
    incomingByte = Serial.read();
    Serial.print("I received: ");
    Serial.println(incomingByte);
    if(incomingByte!=55)
    {
    String thisString = String(incomingByte);
    str=str+thisString;
    }
    else{
      Serial.println(str);
      break;
    }
  }
}
a=str.toInt();
str = "";
Serial.println(a);
int count=0;
Servo1.write(90);
while(1)
{
  if(digitalRead(ProxSensor)==LOW)      //Check the sensor
output
  {
    count=count+1;
    Serial.println(count);
    if (count==a)
    {
         Servo1.write(0);
    digitalWrite(4,HIGH);
   delay(2000);
   digitalWrite(4,LOW);
   break;
    }
delay(1000);
  }
}
   }
  else{
    digitalWrite(3,HIGH);
  delay(2000);
  digitalWrite(3,LOW);
  }
  rfid.PICC_HaltA();
  rfid.PCD_StopCrypto1();


}
```

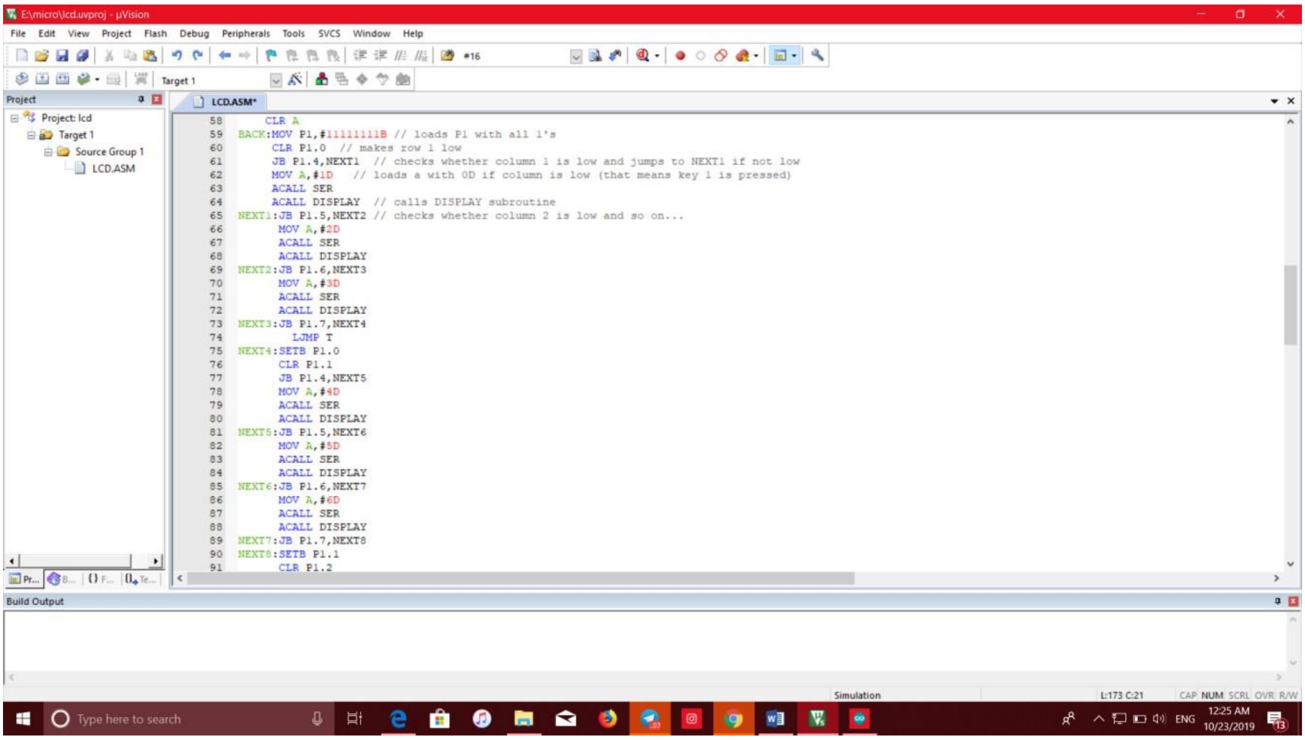
Pin Configurations for Following Commands:



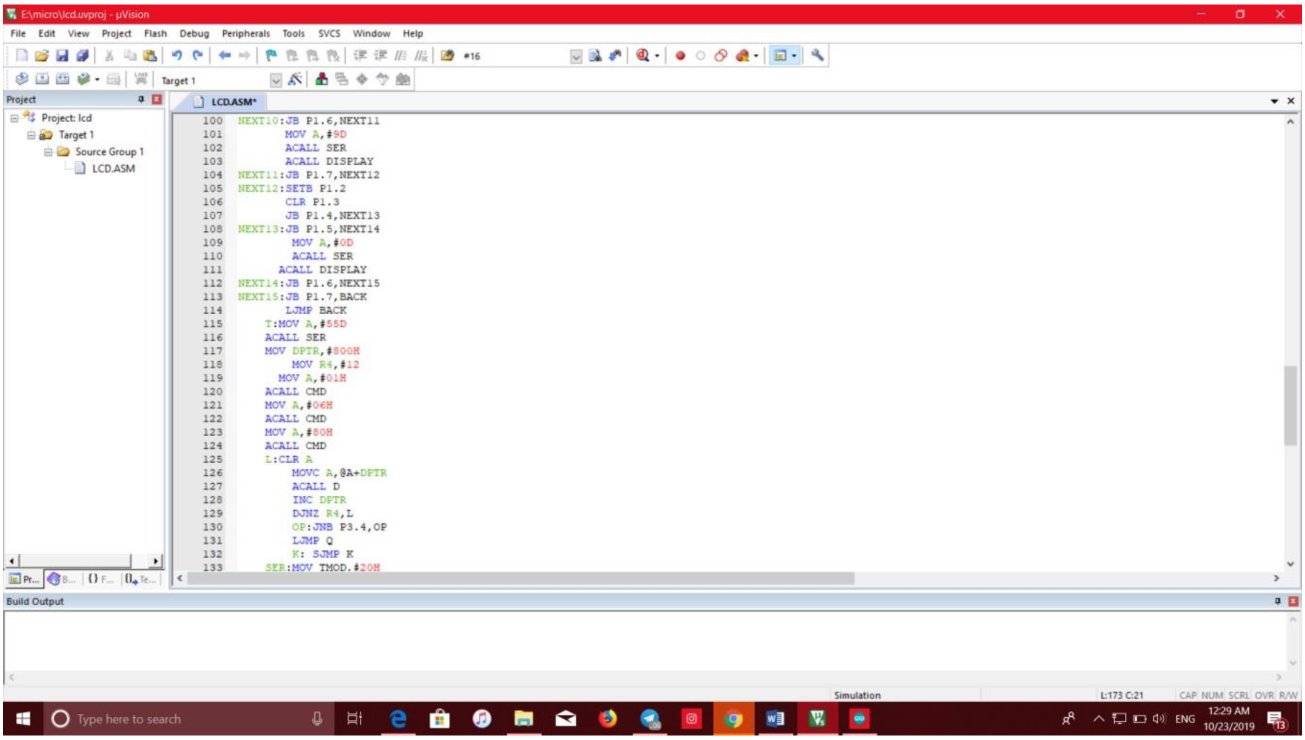
```
ORG 00H
    Q:CLR P3.6
    CLR P3.7
    CLR P3.4
    MOV A,#38H
    ACALL CMD                                 ;To send command to LCD
    MOV A,#0FH                                ;To display on cursor blinking
    ACALL CMD
    MOV A,#01H                                ;clear display
    ACALL CMD
    MOV A,#06H                                ;Increment cursor
    ACALL CMD
    MOV A,#80H                                ;force cursor to 1st line
    ACALL CMD
    MOV DPTR,#400H
    MOV R4,#10
    LOOP:CLR A
    MOVC A,@A+DPTR
    ACALL D
    INC DPTR
    DJNZ R4,LOOP
    ROP:JB P3.6,JUMP                      ;To check Rs5 or Rs.10 card placed
    JB P3.7,JUMP1
    SJMP ROP
    JUMP1:MOV A,#01H
    ACALL CMD
    MOV A,#06H
    ACALL CMD
    MOV A,#80H
    ACALL CMD
    MOV DPTR,#650H
    MOV R4,#12
    LOO:CLR A
    MOVC A,@A+DPTR
```



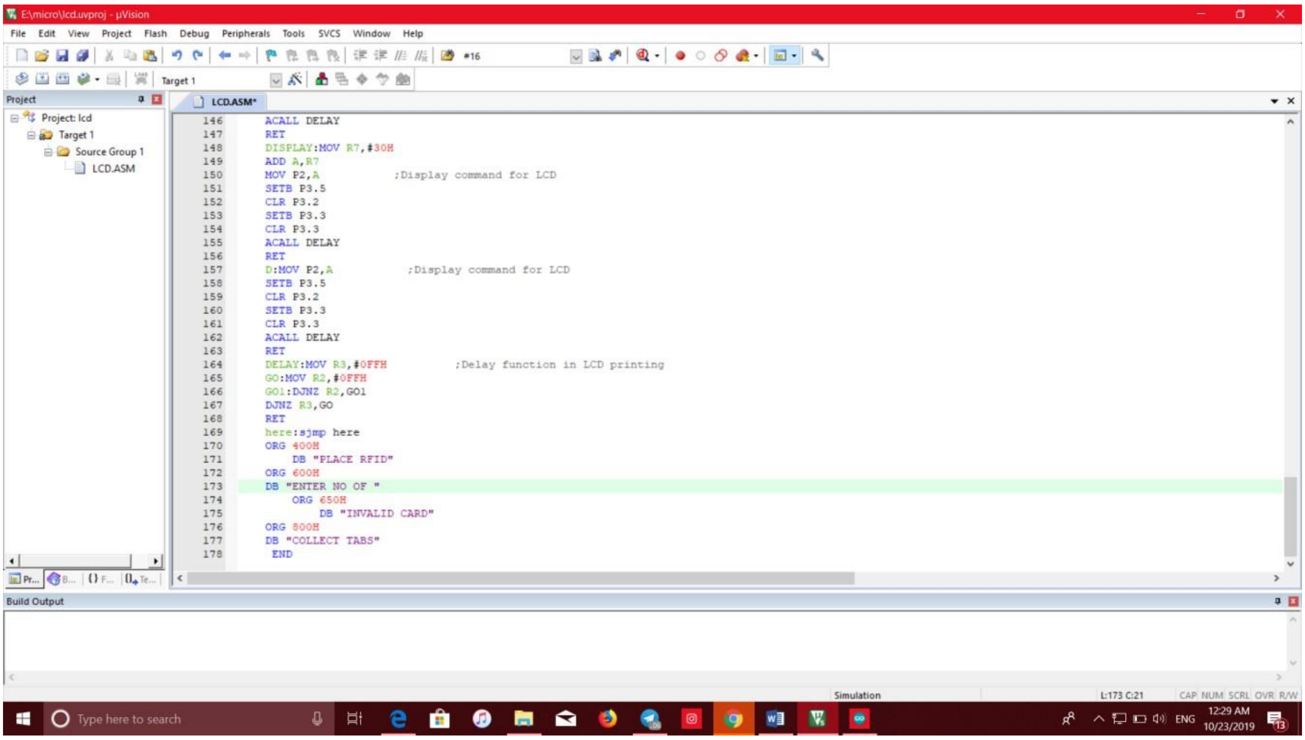
```
    MOV DPTR,#650H
    MOV R4,#12
    LOO:CLR A
    MOVC A,@A+DPTR
    ACALL D
    INC DPTR
    DJNZ R4,LOO
    LJMP Q
    JUMP:MOV A,#01H
    ACALL CMD
    MOV A,#06H
    ACALL CMD
    MOV A,#80H
    ACALL CMD
    MOV DPTR,#600H
    MOV R4,#16
    LO:CLR A
    MOVC A,@A+DPTR
    ACALL D
    INC DPTR
    DJNZ R4,LO
    MOV A,#01H
    ACALL CMD
    MOV A,#06H
    ACALL CMD
    MOV A,#80H
    ACALL CMD
    CLR A
BACK:MOV P1,#11111111B // loads P1 with all 1's
     CLR P1.0  // makes row 1 low
     JB P1.4,NEXT1  // checks whether column 1 is low and jumps to NEXT1 if not low
     MOV A,#1D   // loads a with 0D if column is low (that means key 1 is pressed)
     ACALL SER
     ACALL DISPLAY  // calls DISPLAY subroutine
```



```
    CLR A
BACK:MOV P1,#11111111B // loads P1 with all 1's
     CLR P1.0  // makes row 1 low
     JB P1.4,NEXT1  // checks whether column 1 is low and jumps to NEXT1 if not low
     MOV A,#1D   // loads a with 0D if column is low (that means key 1 is pressed)
     ACALL SER
     ACALL DISPLAY  // calls DISPLAY subroutine
NEXT1:JB P1.5,NEXT2 // checks whether column 2 is low and so on...
      MOV A,#2D
      ACALL SER
      ACALL DISPLAY
NEXT2:JB P1.6,NEXT3
      MOV A,#3D
      ACALL SER
      ACALL DISPLAY
NEXT3:JB P1.7,NEXT4
        LJMP T
NEXT4:SETB P1.0
      CLR P1.1
      JB P1.4,NEXT5
      MOV A,#4D
      ACALL SER
      ACALL DISPLAY
NEXT5:JB P1.5,NEXT6
      MOV A,#5D
      ACALL SER
      ACALL DISPLAY
NEXT6:JB P1.6,NEXT7
      MOV A,#6D
      ACALL SER
      ACALL DISPLAY
NEXT7:JB P1.7,NEXT8
NEXT8:SETB P1.1
      CLR P1.2
```

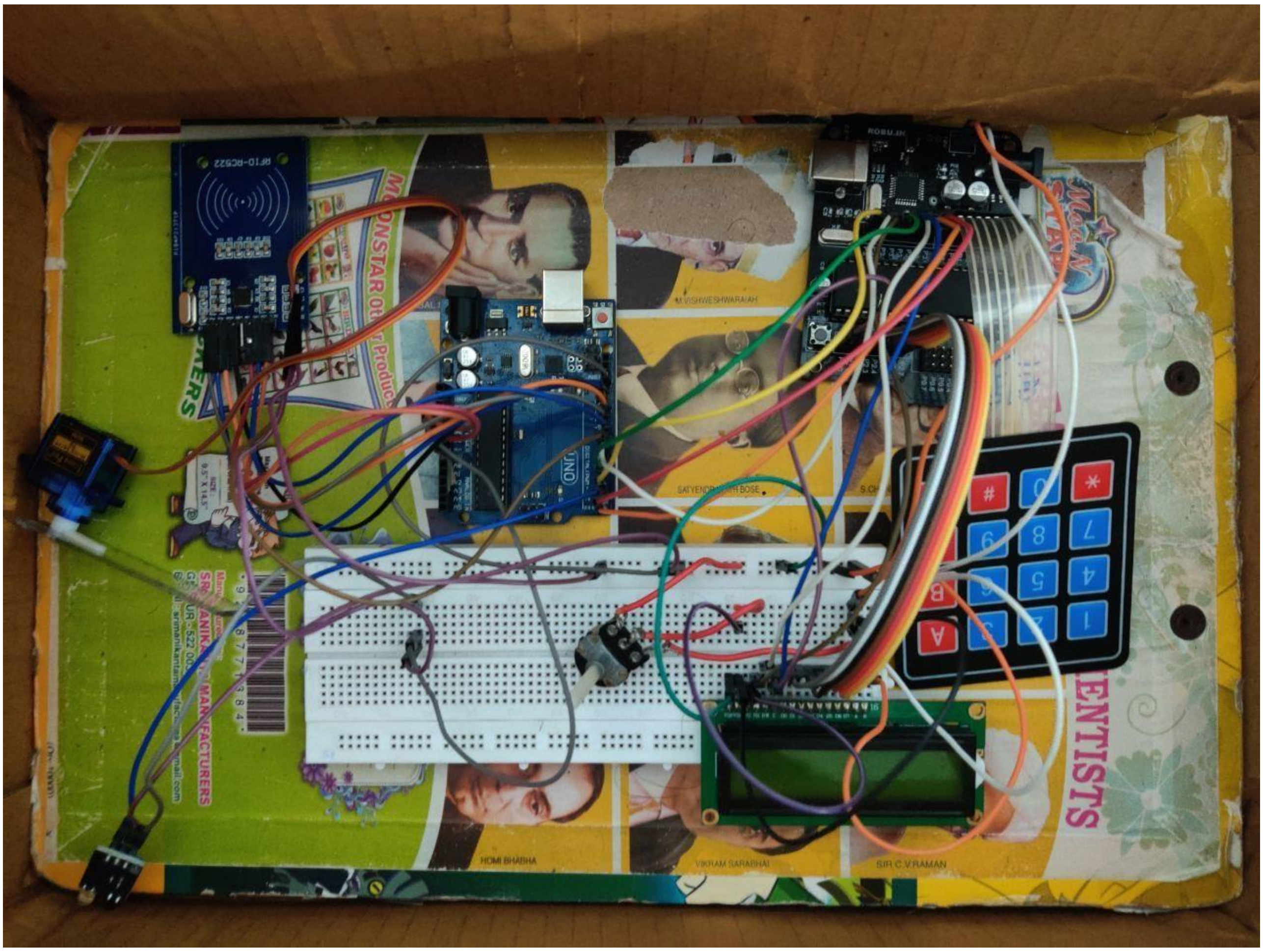


```
NEXT10:JB P1.6,NEXT11
       MOV A,#9D
       ACALL SER
       ACALL DISPLAY
NEXT11:JB P1.7,NEXT12
NEXT12:SETB P1.2
       CLR P1.3
       JB P1.4,NEXT13
NEXT13:JB P1.5,NEXT14
        MOV A,#0D
        ACALL SER
      ACALL DISPLAY
NEXT14:JB P1.6,NEXT15
NEXT15:JB P1.7,BACK
       LJMP BACK
    T:MOV A,#55D
    ACALL SER
    MOV DPTR,#800H
        MOV R4,#12
      MOV A,#01H
    ACALL CMD
    MOV A,#06H
    ACALL CMD
    MOV A,#80H
    ACALL CMD
    L:CLR A
        MOVC A,@A+DPTR
        ACALL D
        INC DPTR
        DJNZ R4,L
        OP:JNB P3.4,OP
        LJMP Q
        K: SJMP K
    SER:MOV TMOD.#20H
```



```
ACALL DELAY
RET
DISPLAY:MOV R7,#30H
ADD A,R7
MOV P2,A            ;Display command for LCD
SETB P3.5
CLR P3.2
SETB P3.3
CLR P3.3
ACALL DELAY
RET
D:MOV P2,A            ;Display command for LCD
SETB P3.5
CLR P3.2
SETB P3.3
CLR P3.3
ACALL DELAY
RET
DELAY:MOV R3,#0FFH          ;Delay function in LCD printing
GO:MOV R2,#0FFH
GO1:DJNZ R2,GO1
DJNZ R3,GO
RET
here:sjmp here
ORG 400H
    DB "PLACE RFID"
ORG 600H
DB "ENTER NO OF "
    ORG 650H
        DB "INVALID CARD"
ORG 800H
DB "COLLECT TABS"
 END
```

Hardware Simulation:



Conclusion:

We successfully implemented the real time application of 8051 as a candy dispenser with RFID reader. The microcontroller 8051 AT89S52 is used as master and Arduino UNO as slave to make the working prototype. The 8051 is programmed entirely in assembly language and Arduino in Arduino IDE.

REFERENCES :

electronicshub.org for 8051 code references

Arduino forum for RFID interface

----------THANK YOU---------