ADVANCED ACADEMIC CENTER

A CENTER FOR INTER-DISCIPLINARY RESEARCH
2018-19

AUTOMAT

SUPERVISED BY

MR YELMA CHETHAN
REDDY

GOKARAJU RANGARAJU
INSTITUTE OF ENGINEERING AND TECHNOLOGY
AUTONOMOUS

Advanced Academic Center

(A Center For Inter-Discilinary Research)

This is to certify that the project titled
“ ---------**AUTOMAT**---------”
is a bonafide work carried out by the following students in partial fulfilment of the requirements for Advanced Academic Center intern, submitted to the chair, AAC during the academic year
2018-2019

NAME	ROLL NO.	BRANCH
K. Harshini	18241A04K5	ECE
Vandana Yalla	18241A05N8	CSE
K Bharat Varma	18241A04P0	ECE

NAME	ROLL N0	BRANCH
S Sathvik	18241A04A2	ECE
G Suhas	18241A04Q0	ECE
S Harsha Veena	18241A04T0	ECE
V Vamshi Krishna	18241A0556	CSE
Rishwanth Perumandla	18241A05S0	CSE
Manda Poojitha	18241A05X1	CSE
U Reshma Shravani	18241A05Z5	CSE
Reddy Richa Vasavi	18241A04S8	ECE
Nampally Lakshmi Prasad	18241A05A0	CSE
T Naveen	18241A04B2	ECE
Kamineni Lokesh	18241A0582	CSE
L Sri Likhitha	18241A0525	CSE
B Jaswanth Kranthi	18241A04P3	CSE

This work was not submitted or published earlier for any study

Mr.Yelma
Chethan Reddy
Project Supervisor

Dr.B.R.K.Reddy
Program Coordinator

Dr.Ramamurthy Suri
Associate Dean,AAC

Gokaraju Rangaraju Institute of Engineering & Technology
Griet
1997

ADVANCED ACADEMIC CENTER

ACKNOWLEDGEMENTS

We express our deep sense of gratitude to our respected Director , Gokaraju Rangaraju Institute of Engineering and Technology for the valuable guidance and for permitting us to carry out this project.

With immense pleasure, we record our deep sense of gratitude to our respected principal, for permitting us to carry out this project.

We are thankful to Associate Dean ,Advance Academic Centre for providing us appropriate ecosystem required for the project to complete.

We are thankful to our project supervisor who spared valuable time for us and influence novel ideas to guide us. I am indebted to all the above without whom I would not have concluded the project.

TABLE OF CONTENTS

ABSTRACT

AUTOMAT is an Arduino based project whose main aim is to provide compact ,unmanned services to it’s users. The Automat is a Vending Machine cum Coffee Maker, the billing transactions take place through an RFID Pre-Paid Card System.

The project is as mentioned previously works on the Arduino and its various applications. The vending machine works on concepts of pulley and rail systems which are carried through stepper motors. For the dispensing unit motors are used. As for the coffee maker the water is suctioned out and milk powder and coffee powder are added and then the coffee is poured out through suctioning. The RFID system is a pre paid card one where it is scanned and then the user can opt according to their wish using a keypad. The system displays the balance and the amount to be cut.

The Automat is cost efficient and it can be modified to adjust to the users needs.

INTRODUCTION

1.1. Purpose of Plan

As most of the work in this area has been done regarding the Arduino & its applications and the mechanical working of pulleys, the challenging part of this project were the rail and pulley systems.

a) More realistic experience to the user.

b) The balance system as a way of transactions

ensures safety.

c) Proper dispensing of milk and the coffee.

1.2. Project objectives

In this project we will be constructing a Vending machine with a coffee making system the mode of transactions will be made through a pre-paid RFID Cards using open source technology Arduino IDE as well as Mechatronics.

1.3. Project goals

- To construct a body for the vending machine and the milk dispensing system.
- To develop an appropriate program in the Arduino microchip to interact with the RFID scanner and the key pad.
- To compile all the developed modules that we constructed above.
- To produce a vending machine plus coffee maker which is compact and cost effective to be used in college departments and offices.

1.4. Scope Definition

The project is limited to designing the Automat and write a program into the Arduino microcontroller. The Vending machine will be limited to 4 chambers and will have weight restrictions. Arduino programs contains instructions mediating between the entry choice in the keypad after, the card is scanned and the pulley system & the pump.

LITERATURE REVIEW

2. Literature review

This literature review explores potential information to identify current knowledge and key issues relating to development of Automat, which are divided into two sections: Arduino interface with the rail-pulley system and the pump & programming and the pre-paid RFID system,

2.1. Arduino

The Arduino was born at the Interaction Design Institute in the city of Ivrea, Italy in 2005. Prof. Massimo Banzi of Interaction Design Institute was looking for a solution for an expensive microprocessor.

In his mind he wanted to make something that is cheap and easy to use and can be able to integrate with the other hardware. So with the help of David Cuartielles, an engineer visiting the Interaction Design Institute from the Malono University, Sweden and with the help of two computer science students they came up with a simple and efficient microcontrollers and they decided to name it after a local bar named Arduino.

As time passed they decided to make Arduino an open source hardware as the Interaction Design Institute was closing down. They feared that their valuable design would be wasted, so they made it public.

With some fee anybody can produce and design a microcontroller and because it’s simple and is easy to install and integrate with other sensors ,modules ,etc made it is very popular.The Arduino microcontroller belongs to the AT-mega microcontroller family.

SOFTWARE AND HARDWARE SPECIFICATION

3.1.Software

- Operating System: Windows 7 (x86 & x64)
- S/W Tool: Arduino IDE

Hardware requirements

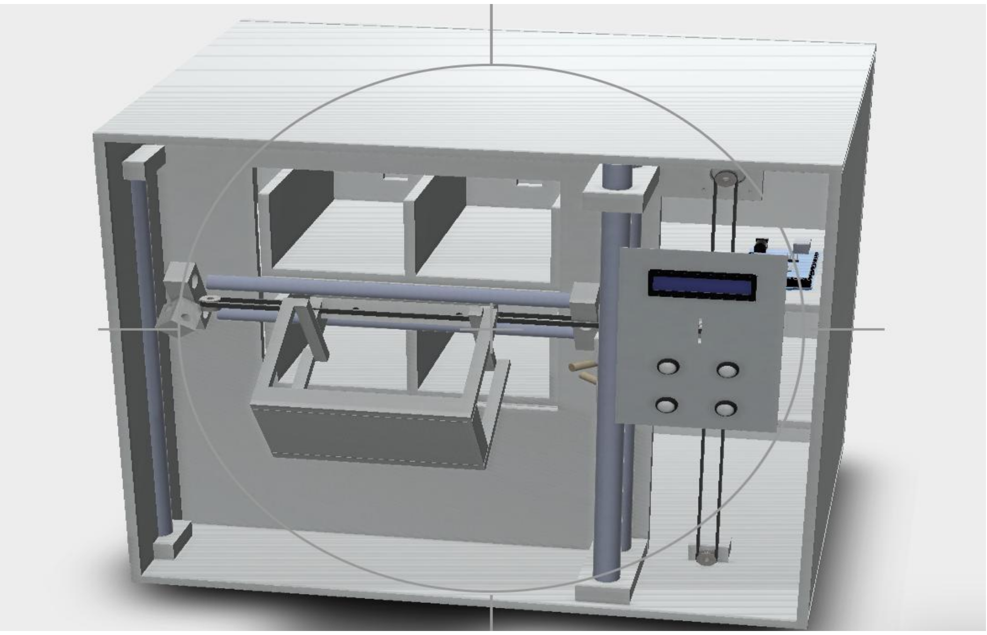
- Arduino Mega
- DC-DC LM2596 Buck Converter
- 16×2 LCD
- 360 Degree Continuous Rotation Servo Motor MG996R
- Stepper Motor NEMA 17
- DC Motor
- A4988 Stepper Motor Driver
- RFID RC522
- Water Pump
- Key Pad
- Micro Limit Switch
- 5-8V Power Source
- 12V Power Source
- Plywood 8mm thickness
- Wood
- Ball Bearings (22mm)
- 3mm Screws
- 5mm Bolts
- PVC pipes (22mm dia)

- Pipes(6mm dia)
- Belts
- Drills
- Hacksaw
- Epoxy
- Shaft Coupler
- Jumper Wires

Project Design And Implementation

The design was made taking into account that the arrangement must be compact. A carrier is included to ensure the quality of products does not get affected. Also the system must be stable to make the coffee without spillage. The design is done such that there are 3 levels to the machine. The topmost level and the middle level have 2 vending spaces each. The last one has space for a heater and a flask. Between the middle and the lowest level there is a gap at the side to insert two containers for the dispensing of coffee powder and milk powder.

The model given below is the design of the vending machine the coffee machine part will be put in between the gaps.



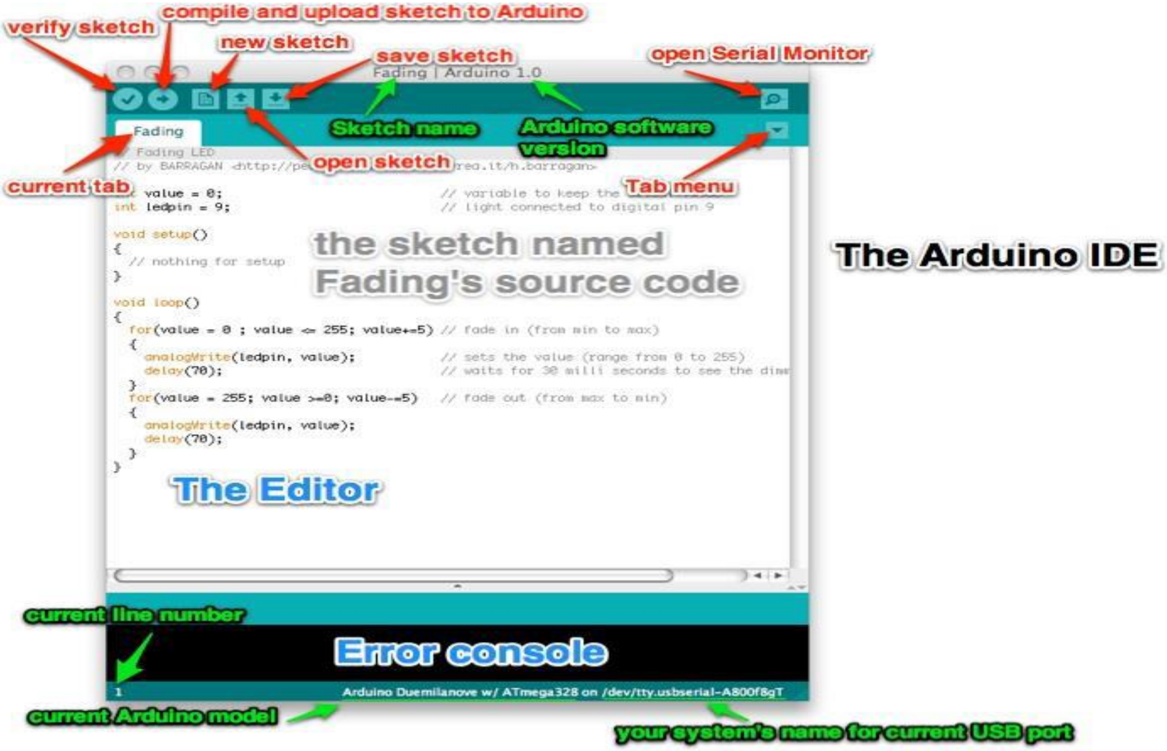
MAIN COMPONENTS

5.1. Arduino IDE

The Arduino integrated development environment (IDE) could be a cross-platform application written in Java, and derives from the IDE for the process programming language and also the Wiring projects. It’s designed to introduce programming to artists and different newcomers unfamiliar with code development. It includes a code editor with options like syntax highlighting, brace matching, and automatic indentation, and is additionally capable of compilation and uploading programs to the board with one click. A program or code written for Arduino is named a "sketch".

Arduino programs are written in C or C++. The Arduino IDE comes with a code library referred to as "Wiring" from the first Wiring project that makes several common input/output operations a lot of easier. The users would like solely to outline 2 functions to create a possible cyclic government program:

- setup (): a function run once at the beginning of a program that may initialize settings
- loop (): a function referred to as repeatedly till the board powers off+ programming languages.



```
Fading LED
// by BARRAGAN <http://pe                rea.it/h.barragan>

  value = 0;                           // variable to keep the
int ledpin = 9;                        // light connected to digital pin 9

void setup()
{
  // nothing for setup
}

void loop()
{
  for(value = 0 ; value <= 255; value+=5) // fade in (from min to max)
  {
    analogWrite(ledpin, value);           // sets the value (range from 0 to 255)
    delay(70);                            // waits for 30 milli seconds to see the dimm
  }
  for(value = 255; value >=0; value-=5)   // fade out (from max to min)
  {
    analogWrite(ledpin, value);
    delay(70);
  }
}
```

Arduino Duemilanove w/ ATmega328 on /dev/tty.usbserial-A800f8gT

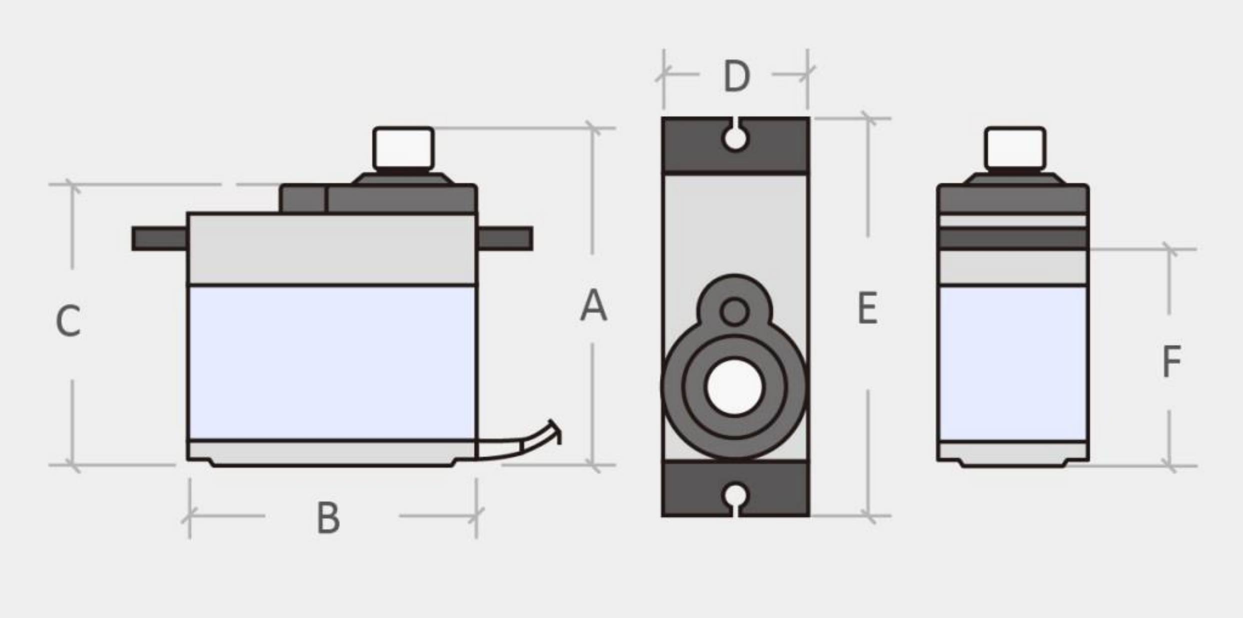
5.2.Arduino Mega

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560. It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega is compatible with most shields designed for the Arduino Duemilanove or Diecimila.

5.3360 Degree Rotation Servo Motor MG996R

It is a digital metal gear servo for Robotic, clockwise/counterclockwise rotation Digital servo.

- **Specification:**
- Weight: 55g
  Dimension: 40.7×19.7×42.9mm
  Stall torque: 9.4kg/cm (4.8v); 11kg/cm (6.0v)
  Operating speed: 0.19sec/60degree (4.8v); 0.15sec/60degree (6.0v)
  Operating voltage: 4.8~ 6.6v
  Gear Type: Metal gear
  Temperature range: 0- 55℃
  Servo Plug: JR (Fits JR and Futaba)
  Dead band width: 1us
  servo wire length: 32cm
  servo arms &screws included and fit with Futaba servo arm
  It’s universal “S” type connector that fits most receivers, including Futaba, JR, Hitec ,GWS, Cirrus, Blue Bird, Blue Arrow, Corona, Berg, Spektrum.
  CE &RoHS approved
  We have upgraded our servo gear set and shaft to aluminum 6061-T6.
  It is stronger and lighter than copper.



5.4. Stepper Motor NEMA 17

NEMA17 4.2Kgcm Stepper Motor for high torque applications. Ideal Motor for ATM machines, 3D printers, Peristaltic pumps and job positioning and rotating applications. Standard NEMA 17 frame size. 4.2Kgcm bipolar configuration motor

Specifications

Step Angle: 1.8°

Number of Phase: 2

Rated Voltage: 3.2V

Rated Current: 1.4A

Resistance per phase: 2E

Inductance per phase: 2.8mH

Holding Torque: 4.2 kgcm

Weight:300 grams

Coil A: Yellow & Red

Coil B: Orange & Brown

Length without shaft: 40mm

Shaft Length: 24mm

Shaft Lenght : D-shaft

Width: 42.3mm

5.5. A4988 stepper motor driver carrier

The A4988 stepper motor driver carrier is a breakout board for Allegro’s easy-to-use A4988microstepping bipolar stepper motor driver and is a drop-in replacement for theA4983stepper motor driver carrier. The driver features adjustable current limiting, overcurrent protection, and five different micro-step resolutions. It operates from 8 – 35V and can deliver-up to 2A per coil.

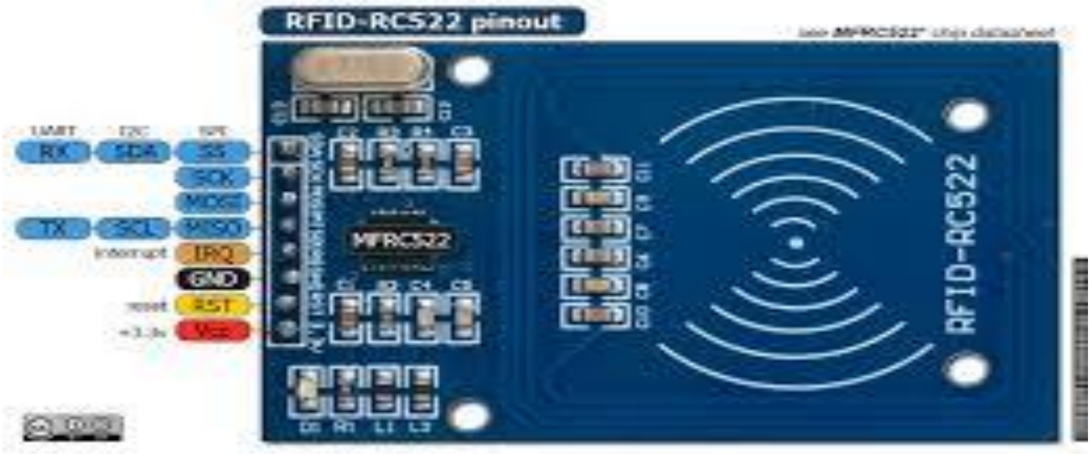
5.7.DC Motor

This is a small DC motor with a small circular shaft of diameter 2mm which measures about 8mm long. The motor has an operating voltage range of 1.5 to 4.5V DC and a no-load speed of 2300 RPM (@4.5VDC, 70mA). The motor can be connected with gear shaft and propeller.

5.6. RFID RC522

RFID or Radio Frequency Identification system consists of two main components, a transponder/tag attached to an object to be identified, and a Transceiver also known as interrogator/Reader.

Frequency Range	13.56 MHz ISM Band
Host Interface	SPI / I2C / UART
Operating Supply Voltage	2.5 V to 3.3 V
Max. Operating Current	13-26mA
Min. Current(Power down)	10µA
Logic Inputs	5V Tolerant
Read Range	5 cm



5.7. Water Pump

The DC6-12V MINI Aquarium water Pump R385 is the perfect choice for any project that requires water to be moved from one place to another. The pump can handle pumping heated liquids up to a temperature of 80°C and when suitably powered can suck water through the tube from up to 2m and pump water vertically for up to 3m.

Specifications

Recommended voltage ratings are 9v 1A or 12v 1A.

Maximum suck range of 2m.

The maximum head range of 3m.

Works with liquids of up to 80°C.

The maximum flow rate of up to 1 – 3L/min.

CONSTRUCTION

The vending machine features four discharging units controlled via four continuous rotation servo motors, a carrier system controlled via stepper motors, an LCD, a keppad for selecting an item and a RFID Scanner.

6.1Building the automat structure

- It is done by cutting the 8 mm thick Plywood board to size. For cutting the plwood a hacksaw was used. After cutting all panels using the saw .Then making the openings in some of the panels using the hacksaw.
- The panels are Assembled using some wood glue and screws. For fastening the panels 90 degrees angle clamps were used. And hammer the 3mm nails in.

6.2.Rail-Pulley System

For the rail-pulley system after cutting the PVC tubes to size using a hand saw. The diameter of the tube for the horizontal rail is 16 mm, while for the vertical rail the diameter is 20 mm. Then 18mm slots are made on a piece of wood and the pipes ate fastened to it.

- The horizontal rail is made out of two 27 cm long tubes, while the vertical rail is made out of three 45 cm long tubes.
- Next are the sliders made by using 21 by 21 cm wood board on which 8 mm holes are made.
- Then 8 mm threaded rods are inserted through these holes and using washers and nuts secured bythe 22 mm bearings. As for the horizontal slider, the same method is used.
- Then the the stepper motor for the horizontal movement is attached. First,the motor is fastened on an 8 mm plywood board, a supporting piece of wood is added to it and also the slotted part is secured to it.Finally the whole assembly is attached.
- The same is done to the horizontal slider but the additional wood board is attached using some epoxy in the rail slots to make the whole rail system stiffer.

- Every component is attatched.
- For the horizontal timing belt. The length is measured and then it is secured to the slider using a zip tie. The stepper motor is attached on the top of the machine using a piece of wood and some bolts. At the bottom the pulley is attached and in a similar way installed the timing belt.

6.3.Discharge units

- Made with a helical coil out of 3 mm thick metal wire by wrapping it around a 5 cm in diameter spray paint can. After that using a glue gun it is secured to a continuous rotation servo motor.

6.4.The milk dispenser

The hot water flask is drilled into and the water pump and the tubes are secured.

6.5**.The coffee and milk powder dispenser**

The coffee and milk powder flasks are placed in position and the bottom of the flasks are drilled into.

6.6.The Keypad and RFID Scanner and Panel

Finally the closing panel is placed and the Keypad,LCD Display and the RFID scanner are attached to it.

CODE

Code for the RFID Pre-Paid Transaction System ,the Keypad and the Pulley Motors

```
#include <SPI.h>
#include <MFRC522.h>
#include <Keypad.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#define RST_PIN         5          // Configurable, see typical pin layout above
#define SS_PIN          53         // Configurable, see typical pin layout above
#include <Servo.h>
Servo myservo;
LiquidCrystal_I2C lcd(0x27, 16, 2);
MFRC522 mfrc522(SS_PIN, RST_PIN);   // Create MFRC522 instance
char ch;
const int stepPin = 42;
const int dirPin = 44;
const int stepPin1 = 38;
const int dirPin1 = 40;
const byte ROWS = 4; //four rows
const byte COLS = 4; //four columns
//define the cymbols on the buttons of the keypads
char hexaKeys[ROWS][COLS] = {
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};
byte rowPins[ROWS] = {9,8,7,6}; //connect to the row pinouts of the keypad
byte colPins[COLS] = {10,4,3,2}; //connect to the column pinouts of the keypad
Keypad customKeypad = Keypad( makeKeymap(hexaKeys), rowPins, colPins, ROWS, COLS);
```

Keypad setup

```
void setup() {
  // Sets the two pins as Outputs
  pinMode(stepPin,OUTPUT);
  pinMode(dirPin,OUTPUT);
  pinMode(stepPin1,OUTPUT);
  pinMode(dirPin1,OUTPUT);
  Serial.begin(9600);
   SPI.begin();
  mfrc522.PCD_Init();
  lcd.begin();
  lcd.backlight();
    lcd.clear();
  lcd.setCursor(1,1);
  lcd.print("WELCOME");
  myservo.attach(20);
}
void loop() {
    char customKey = customKeypad.getKey();

  switch(customKey)
  {
   case '1': buy_item();
              break;
    case '2': motor();
              break;
   case 'A': check_balance();
              break;
  }
}
```

```
void buy_item()
{
    Serial.println(F("Read personal data on a MIFARE PICC:"));
//shows in serial that it is ready to read
  // Prepare key - all keys are set to FFFFFFFFFFFFh at chip
delivery from the factory.
  MFRC522::MIFARE_Key key;
  for (byte i = 0; i < 6; i++) key.keyByte[i] = 0xFF;

  //some variables we need
  byte block;
byte len;
  MFRC522::StatusCode status;
  //-------------------------------------------

  // Look for new cards
  if ( ! mfrc522.PICC_IsNewCardPresent()) {
    return;
  }

  // Select one of the cards
  if ( ! mfrc522.PICC_ReadCardSerial()) {
    return;
  }

  Serial.println(F("**Card Detected:**"));

  //-------------------------------------------

  mfrc522.PICC_DumpDetailsToSerial(&(mfrc522.uid)); //dump
some details about the card
```

```
 //mfrc522.PICC_DumpToSerial(&(mfrc522.uid));      //uncomment this to
see all blocks in hex


  //------------------------------------------


  //Serial.print(F("Name: "));
  byte buffer[38];
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////
  byte buffer1[18];
  block = 1;
len = 18;


  //------------------------------------------ GET FIRST NAME
  status =
mfrc522.PCD_Authenticate(MFRC522::PICC_CMD_MF_AUTH_KEY_
A, 1, &key, &(mfrc522.uid)); //line 834 of MFRC522.cpp file
  if (status != MFRC522::STATUS_OK) {
    Serial.print(F("Authentication failed: "));
    Serial.println(mfrc522.GetStatusCodeName(status));
    return;
  }
   status = mfrc522.MIFARE_Read(block, buffer1, &len);
  if (status != MFRC522::STATUS_OK) {
    Serial.print(F("Reading failed: "));
    Serial.println(mfrc522.GetStatusCodeName(status));
    return;
  }

//PRINT FIRST NAME
 String ch[20];
Serial.println("Card Holder Name :");
```

```
 Serial.println(" ");
  lcd.clear();
 for (uint8_t i = 0,j=0; i < 16; i++,j++)
 {
 if (buffer1[i] != 32)
  {
  lcd.setCursor(j,0);
  Serial.println(i);
  lcd.write(buffer1[i]);
  }
  Serial.println(" ");
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//////////////////////////////////////////////////////
  byte buffer2[18];
block = 2;
  status =
mfrc522.PCD_Authenticate(MFRC522::PICC_CMD_MF_AUTH_KEY_A, 2,
&key, &(mfrc522.uid)); //line 834
  if (status != MFRC522::STATUS_OK) {
  Serial.print(F("Authentication failed: "));
  Serial.println(mfrc522.GetStatusCodeName(status));
  return;
 }
 status = mfrc522.MIFARE_Read(block, buffer2, &len);
 if (status != MFRC522::STATUS_OK) {
 Serial.print(F("Reading failed: "));
 Serial.println(mfrc522.GetStatusCodeName(status));
 return;  }
 Serial.println("Amount Present :");
  for (uint8_t i = 0; i < 1; i++) {
    Serial.println(buffer2[i]);
  }
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
Serial.setTimeout(10000L) ;
  if(buffer2[0]>=10)
  {  byte cost = 10;
    Serial.println(F("Amount Left : "));
    len = buffer2[0]-cost;
    Serial.println(len);
    lcd.setCursor(0,1);
    lcd.print("Cost:");
    lcd.print(cost);
    buffer[0]=len;
    for (byte i = len; i < 30; i++) buffer[i] = ' ';     // pad with spaces
    block = 2;
    //Serial.println(F("Authenticating using key A..."));
    status =
mfrc522.PCD_Authenticate(MFRC522::PICC_CMD_MF_AUTH_KEY_A,
block, &key, &(mfrc522.uid));
    if (status != MFRC522::STATUS_OK) {
      Serial.print(F("PCD_Authenticate() failed: "));
      Serial.println(mfrc522.GetStatusCodeName(status));
      return;
   }
  // Write block
    status = mfrc522.MIFARE_Write(block, buffer, 16);
   if (status != MFRC522::STATUS_OK) {
   Serial.print(F("MIFARE_Write() failed: "));
 Serial.println(mfrc522.GetStatusCodeName(status));
    return;
    }
    else
    Serial.println(F("MIFARE_Write() success: "));
  Serial.println(F("\n**End Reading**\n"));
  delay(1000); //change value if you want to read cards faster
  mfrc522.PICC_HaltA();
  mfrc522.PCD_StopCrypto1();
   motor();}
  else
  {
     Serial.println(F("\n**End Reading**\n"));
delay(1000); //change value if you want to read cards faster
  mfrc522.PICC_HaltA();
  mfrc522.PCD_StopCrypto1();
  }
  Serial.println(F("\n**End Reading**\n"));
  delay(3000);
  lcd.clear();
  lcd.print("WELCOME");
  }
//*****************************************************************
void motor(){
 digitalWrite(dirPin,LOW);
digitalWrite(dirPin1,HIGH);
```

```
for(int x = 0; x < 10000 ;x++) {
    digitalWrite(stepPin,HIGH);
    delayMicroseconds(500);
    digitalWrite(stepPin,LOW);
    delayMicroseconds(500);
   //digitalWrite(stepPin1,HIGH);
   //delayMicroseconds(500);
   //digitalWrite(stepPin1,LOW);
   //delayMicroseconds(500);
   }
 }
void check_balance(){
      Serial.println(F("Read personal data on a MIFARE PICC:"));    //shows in serial that it is ready to read
  // Prepare key - all keys are set to FFFFFFFFFFFFh at chip delivery from the factory.
  MFRC522::MIFARE_Key key;
  for (byte i = 0; i < 6; i++) key.keyByte[i] = 0xFF;
   //some variables we need
  byte block;
  byte len;
  MFRC522::StatusCode status;
//-------------------------------------------
// Look for new cards
  if ( ! mfrc522.PICC_IsNewCardPresent()) {
   return;
}
```

```
 // Select one of the cards
  if ( ! mfrc522.PICC_ReadCardSerial()) {
    return;
 }
  Serial.println(F("**Card Detected:**"));
  //--------------------------------------------
  mfrc522.PICC_DumpDetailsToSerial(&(mfrc522.uid)); //dump some details about the card
//mfrc522.PICC_DumpToSerial(&(mfrc522.uid));      //uncomment this to see all blocks in hex
 //--------------------------------------------
//Serial.print(F("Name: "));
  byte buffer[38];
/////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
  byte buffer1[18];
   block = 1;
  len = 18;
  //-------------------------------------------- GET FIRST NAME
  status = mfrc522.PCD_Authenticate(MFRC522::PICC_CMD_MF_AUTH_KEY_A, 1, &key, &(mfrc522.uid)); //line 834 of MFRC522.cpp file
  if (status != MFRC522::STATUS_OK) {
    Serial.print(F("Authentication failed: "));
    Serial.println(mfrc522.GetStatusCodeName(status));
 return;
}
```

```
status = mfrc522.MIFARE_Read(block, buffer1, &len);
  if (status != MFRC522::STATUS_OK) {
    Serial.print(F("Reading failed: "));
    Serial.println(mfrc522.GetStatusCodeName(status));
   return;
}
  //PRINT FIRST NAME
 Serial.println("Card Holder Name :");
 Serial.println(" ");
 lcd.clear();
 for (uint8_t i = 0,j=0; i < 16; i++,j++)
 {
 if (buffer1[i] != 32)
 {
  lcd.setCursor(j,0);
  Serial.println(i);
  lcd.write(buffer1[i]);
 }
}
  Serial.println(" ");
//////////////////////////////////////////////////////////////////////////////////////////////////////////////
  byte buffer2[18];
  block = 2;
  status =
mfrc522.PCD_Authenticate(MFRC522::PICC_CMD_MF_AUTH_KEY_A, 2,
&key, &(mfrc522.uid)); //line 834
 if (status != MFRC522::STATUS_OK) {
 Serial.print(F("Authentication failed: "));
    Serial.println(mfrc522.GetStatusCodeName(status));
    return;
  }
  status = mfrc522.MIFARE_Read(block, buffer2, &len);
  if (status != MFRC522::STATUS_OK) {
Serial.print(F("Reading failed: "));
Serial.println(mfrc522.GetStatusCodeName(status));
    return;
}
  Serial.println("Amount Present :");
  for (uint8_t i = 0; i < 1; i++) {
    Serial.println(buffer2[i]);
  }
  lcd.setCursor(0,1);
  lcd.print("Balance:");
  lcd.print(buffer2[0]);
  Serial.println(F("\n**End Reading**\n"));
  delay(1000); //change value if you want to read cards faster
  mfrc522.PICC_HaltA();
  mfrc522.PCD_StopCrypto1();
  delay(3000);
  lcd.clear();
  lcd.print("WELCOME");
}
```

CONCLUSION

The Automat works well and the separate modules on it work excellently. All the modules have been run through test cases several times. The RFID system has been made secured after several test runs after which modifications were made. The pulley system has been tested several times and works smoothly. A lot of deigns for the coffee maker were made out of which the most optimum one was executed in the end . The whole project as a whole has shown satisfactory results . It is cost efficient and also very flexibile.

The team has been exposed to many facets of the Arduino and also have basic knowledge of the mechatronics subject.The whole project required a lot of research to come up with efficient designs & code and low cost materials.