# ECE4002- ADVANCED MICROCONTROLLERS

# J COMPONENT REPORT

# TITLE:

# Library Management System using ATMEGA16

## SLOT: E1+TE1

## FACULTY: Prof. GERARDINE


***Submitted by***

***S Kshipra Prasadh 16BIS0135***

***Vaidyanathan 17BIS0135***

***Chandramouli 17BIS0148***


# FALL SEMESTER 2019-20

# SCHOOL OF ELECTRONICS ENGINEERING

# VIT UNIVERSITY, VELLORE

# CERTIFICATE

**This is to certify that S Kshipra Prasadh, 16BIS0135, Vaidyanathan , 17BIS0135, Chandramouli, 17BIS0148, have completed the project, 'Library Management System using ATMEGA16', under my supervision for the course, Advanced Microcontrollers (ECE4002), in the Fall semester 2019-20.**

_____

***(Faculty Signature)***

# ACKNOWLEDGEMENTS

**We would like to thank our professor, Prof. Gerardine for guiding us throughout the project. We are also grateful to VIT University, Vellore, for providing us a platform to present this project.**

***S Kshipra Prasadh 16BIS0135***

***Chandramouli 17BIS0148***

***Vaidyanathan 17BIS0135***

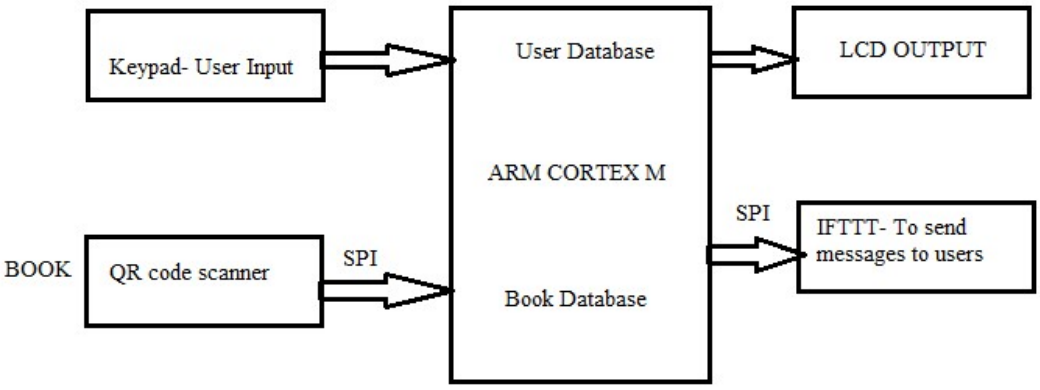
# ABSTRACT

Nowadays, the scale of library is expanding and the quantity of books is increasingly large. Besides, with the growing popularity of open shelf reading the phenomenon of misplaced books is becoming more&more common. Thus, readers finding books and library administrators checking books are also becoming increasingly difficult. The misplaced books can be identified by using a code attached on the shelf. The position of the corresponding book will be displayed on the screen. The data transfer takes place with the help of AMTP technology. However, what has been used most widely now is the library management system based on barcode, whose operation process is beyond complicated. What's more, the barcode has a short life and breaks easily,which greatly influences the efficiency of the library management.

# AIM AND BLOCK DIAGRAM

- The library management system consists of the following functionalities
- Create a database of books available and registered users
- Allow registered users to borrow, reborrow and return books
- Calculate fines for late return of books
- Issue messages to users to return books, if it extends a certain time period.

All these functionalities will be implemented via ATMEGA16




\

# RELATED WORKS

In library, the management of books is very complication and timing consuming. The location of books could be altered by librarian, students, teachers and anyone around the library .Therefore, allocating a book is not an easy task in big library. Indoor positioning is an important technology to help storage management and customer services providing . RFID provides a good wireless platform to facilitate indoor positioning . The problems & issues faced in the library environment like locating the mis-placed or mis-shelved book or materials, reducing the manual work & ease access of the books are done and a solution is developed that could overcome these problems with the better enhanced work .

An RFID Gen2 reader on a moving cart can offer tag reading rates close to 100%, provided that multiplexing (and not splitting) is used at the antennas connected to the reader, by exploiting the changes (due to mobility) of the wireless propagation channel. Library RFID systems could largely benefit of such cost-effective approach, given that only a single reader is employed. A 13.56 MHz RFID system for the management of the library is used .We evaluate the influence that papers or other RFID tags give to the resonant frequency of an RFID tag .

An Image matching system is adopted together with RSS based RFID positioning method to improve the accuracy and robustness of the book positioning system. ,

The first step is to locate the bookshelf containing the book by neural network based RFID. This is to narrow down the area for later image matching to enhance speed and efficiency. The process of borrowing, returning and other details with regards to the student and book, the sequence done at admin desk is carried out in this process of implementation. For which the compatible handheld reader Motorola MC 9090 is used. The database used, is the internal database.

The user when comes to the administration desk, the user need not require help from Library staffs that the user reads the tag present in the ID card of the user and proceed according to the user's requirement like renewal, borrow or return. The image of the user and details are made to display through the user interface. The tools used are like the Visual Studio 2008 in the platform of .NET framework.

he RFID tag is attached on goods such as books. In this case, the resonant frequency of RFID tag is changed by goods. The resonant frequency of the tag attached to a book is not the same compared with the case of tag only. As the volume of book increases, the resonant frequency seems to be low. Normally, the goods act as a parasitic capacitance for the RFID tag and If other RFID Tags are near the target RFID tag, the resonant frequency of target tag is changed. For example, the situation that two pieces of tags are adjacent occurs when thin books are stood on the book shelf. This influence is large compared with the influence of other goods such as a book and, the relation between the resonant frequency and the distance between two pieces of tags is evaluated. Figure 10 shows the results of this evaluation. It is clearly shown that as the distance between two pieces of tags becomes narrow, the resonant frequency decreases greatly. On the other

hand, the combination between the tags is small if the distance of two tags is separated more than 7 cm.

The common problem of mis-placed & mis-shelved searching of the books & materials can be very easily reduced and also there is no need of arranging the books in the order that any book can be placed anywhere inside the library and the same can be located.

# HARDWARE AND SOFTWARE COMPONENTS USED

| HARWARE | SPECIFICATION |
|---|---|
| AVR MICROCONTOLLER | ATMEGA 16 |
| LCD | 16X2 |
| KEYPAD | - |
| ESP8266 | WIFI MODULE |

LCD

- The LCD performs the following functions
- Displays the user menu- for borrow, return and show fine
- Displays error messages- if the user wants to borrow more than 4 books
- Displays fines on users request

KEYPAD

- The keypad is used to
- Allow the user to enter his/her choice to borrow/ return books/ see fines
- To allow the user to enter the user ID for borrowing or returning
- To allow the ADMIN to delete any book record if the book is damaged

Borrow()

- The borrow function is used to allow the user to borrow books
- It checks if the user has already borrowed 4 books
- If not, it links the book id and user id
- Return()
  The return function de-links the book id and user id.
- It also calculates the fine based on the number of days delayed in returning the book
- The result is displayed on an LCD

- There is also a function to delete the books, which can be used by the ADMIN, to delete damaged books

# METHODOLOGY AND CODE

The microcontroller is used to obtain user inputs through keypad. Based on any pending fines, the wifi module is activated, and an email will be sent to the user, using SMTP protocol.

```
/*/*
 * GccApplication2.c
 *
 * Created: 9/11/2019 12:11:42 PM
 * Author : Chandramaouli S
 */
#include <avr/io.h>
#include <time.h>
#include <util/delay.h>
#include <avr/pgmspace.h>
#include <string.h>
#include <stdlib.h>
#define F_CPU 8000000UL
#define KEY_PRT PORTC
#define KEY_DDR DDRC
#define KEY_PIN PINC
#define LCD_Data_Dir DDRB       /* Define LCD data port direction */
#define LCD_Command_Dir DDRA       /* Define LCD command port direction
register */
#define LCD_Data_Port PORTB     /* Define LCD data port */
#define LCD_Command_Port PORTA     /* Define LCD data port */
#define RS PA0              /* Define Register Select (data/command
reg.)pin */
#define RW PA1              /* Define Read/Write signal pin */
#define EN PA2              /* Define Enable signal pin */
#define BAUD_PRESCALE (((F_CPU / (USART_BAUDRATE * 16UL))) - 1)     //
convert to baudrate dec value
unsigned char keypad[4][4] =
{{'7','8','9','A'},{'4','5','6','B'},{'1','2','3','C'},{'
','0','D','E'}};
unsigned int fine=0;
void UART_init(long USART_BAUDRATE)
{
UCSRB |= (1 << RXEN) | (1 << TXEN);         // Turn on transmission and
reception by setting RX Tx bits
UCSRC |= (1 << URSEL) | (1 << UCSZ0) | (1 << UCSZ1);    // Use 8-bit
character sizes
UBRRL = BAUD_PRESCALE; // Load lower 8-bits of the baud rate value
UBRRH = (BAUD_PRESCALE >> 8); // Load upper 8-bits of the baud rate value
}
void UART_TxChar(char c)
{
while (! (UCSRA & (1<<UDRE))); // Wait for empty transmit buffer
UDR = c;
}

void UART_sendString(char *str1)
{
unsigned char s=0;
```

```
while (str1[s]!=0) // string till null
{
UART_TxChar(str1[s]);    // send s to UART_TxChar(s) function
s++;
}
}

char keyboard()
{
    unsigned char colloc, rowloc;
    while(1)
    {KEY_DDR =0xF0;
        KEY_PRT = 0xFF;
        do
        {
            KEY_PRT  &= 0x0F;
            colloc = (KEY_PIN & 0x0F);
        }while(colloc !=0x0F);
        do
        {
            do
            {
                _delay_ms(20);
                colloc = (KEY_PIN & 0x0F);
            } while (colloc == 0x0F);

            _delay_ms(20);
            colloc = (KEY_PIN & 0x0F);
        }while(colloc ==0x0F);
        KEY_PRT = 0xEF;
        asm("NOP");
        colloc = (KEY_PIN & 0x0F);
        if(colloc != 0x0F)
        {
            rowloc=0;
            break;
        }
        KEY_PRT = 0xDF;
        asm("NOP");
        colloc = (KEY_PIN & 0x0F);
        if(colloc != 0x0F)
        {
            rowloc=1;
            break;
        }
        KEY_PRT = 0xBF;
        asm("NOP");
        colloc = (KEY_PIN & 0x0F);
        if(colloc != 0x0F)
        {
            rowloc=2;
            break;
        }
        KEY_PRT = 0x7F;
        asm("NOP");
        colloc = (KEY_PIN & 0x0F);
```

```
        if(colloc != 0x0F)
        {
            rowloc = 3;
            break;
        }
    }
    if(colloc == 0x0E)
    return(keypad[rowloc][0]);
    else if(colloc == 0x0D)
    return(keypad[rowloc][1]);
    else if(colloc == 0x0B)
    return(keypad[rowloc][2]);
    else
    return(keypad[rowloc][3]);
}
void LCD_Command(unsigned char cmnd)
{
    LCD_Data_Port= cmnd;
    LCD_Command_Port &= ~(1<<RS);   /* RS=0 command reg. */
    LCD_Command_Port &= ~(1<<RW);   /* RW=0 Write operation */
    LCD_Command_Port |= (1<<EN);    /* Enable pulse */
    _delay_us(1);
    LCD_Command_Port &= ~(1<<EN);
    _delay_ms(3);
}

void LCD_Char (unsigned char char_data)   /* LCD data write function */
{
    LCD_Data_Port= char_data;
    LCD_Command_Port |= (1<<RS);    /* RS=1 Data reg. */
    LCD_Command_Port &= ~(1<<RW);   /* RW=0 write operation */
    LCD_Command_Port |= (1<<EN);    /* Enable Pulse */
    _delay_us(1);
    LCD_Command_Port &= ~(1<<EN);
    _delay_ms(1);
}

void LCD_Init (void)          /* LCD Initialize function */
{
    LCD_Command_Dir = 0xFF;      /* Make LCD command port direction as o/p
*/
    LCD_Data_Dir = 0xFF;     /* Make LCD data port direction as o/p */
    _delay_ms(20);           /* LCD Power ON delay always >15ms */

    LCD_Command (0x38);      /* Initialization of 16X2 LCD in 8bit mode
*/
    LCD_Command (0x0C);      /* Display ON Cursor OFF */
    LCD_Command (0x06);      /* Auto Increment cursor */
    LCD_Command (0x01);      /* Clear display */
    LCD_Command (0x80);      /* Cursor at home position */
}

void LCD_String (char *str)     /* Send string to LCD function */
{

    int i;
```

```
    for(i=0;str[i]!=0;i++)       /* Send each char of string till the NULL
*/
    {
        LCD_Char (str[i]);
    }
}

void LCD_String_xy (char row, char pos, char *str)/* Send string to LCD
with xy position */
{
    if (row == 0 && pos<16)
    LCD_Command((pos & 0x0F)|0x80); /* Command of first row and required
position<16 */
    else if (row == 1 && pos<16)
    LCD_Command((pos & 0x0F)|0xC0); /* Command of first row and required
position<16 */
    LCD_String(str);     /* Call LCD string function */
}

void LCD_Clear()
{
    LCD_Command (0x01);      /* clear display */
    LCD_Command (0x80);      /* cursor at home position */
}
typedef struct book {
    char ** bookname;
    int book_id;
    int d;
    int m;
    int y;
    int user_id;
    struct data_t * next;
    struct data_t * prev;
} data_t;

data_t * list_base = 0, *current, *previous;

void add_record(char * str1, int num_strings,int d,int m,int y, int l)
{
    if (list_base == 0) {
        list_base = (data_t *) calloc( sizeof(data_t), 1 );
        current = list_base;
    }
    else {
        previous->next = (data_t *) calloc( sizeof(data_t), 1 );
        current = previous->next;
    }
    current->book_id = num_strings;
    current->bookname = calloc(1, strlen(str1));
    strcpy(
    current->bookname, str1);
    current->d=d;
    current->m=m;
    current->y=y;
    current->user_id = l;
    current->next = 0;
```

```
    current->prev = previous;
    previous = current;
}

void borrow(int m, int l) __attribute__((noinline));
void borrow(int m, int l) {
    data_t * check = list_base;
    unsigned int num=0;

    while(check)
    {
        if(check->user_id==l)
        num=num+1;
        check=check->next;
    }
    check = list_base;
    if (num<4) {
        while (check) {
            if (check->book_id == m) {
                check->user_id = l;
                time_t now;
                struct tm *local=localtime(&now);
                check->d=local->tm_mday;
                check->m=local->tm_mon+1;
                check->y=local->tm_year+1900;

            }
            check = check->next;
        }
    }
    else
    {
        LCD_Init();
        LCD_String("max 4 books");
    }
}

void return_book(int c)
{
    data_t * check = list_base;

    while(check)
    {
        if(check->book_id==c)
        {
            check->user_id=NULL;
            time_t now;
            struct tm *local=localtime(&now);
            int d1=local->tm_mday;
            int m1=local->tm_mon+1;
            int y1=local->tm_year+1900;
            int rtd=check->d+14;
            int rtm=check->m;
            char *p;
            if(rtd>31)
            {rtd-=31;
```

```
                rtm++;
            }
            if(d1>rtd&&m1>rtm)
            {
                fine=(d1-rtd)*2+(m1-rtm)*60;
            }
            else if(m1>rtm && d1<rtd)
            {

                fine=(d1+31-rtd)*2;
            }
            if(fine==0)
            {
                LCD_Init();
                LCD_String("no fine");
            }
            else
            {
                itoa(fine,p,10);
                LCD_Init();
                LCD_String(p);
            }
        }
        check=check->next;
    }
}
void delete_record(data_t * rec) __attribute__((noinline));
void delete_record(data_t * rec) {
    data_t * prev, *next;
    int i = 0;
    // get the pointers to N-1 and N+1 (may be 0)
    next = rec->next;
    prev = rec->prev;
    prev->next = next;
    next->prev = prev;
    free(rec->bookname);
}
void LCD_Menu()
{
    LCD_Init();
    LCD_String("1BORROW 2RETURN");  /* write string on 1st line of LCD*/
    LCD_Command(0xC0);       /* Go to 2nd line*/
    LCD_String("3REMOVE");   /* Write string on 2nd line*/
}

int main(void) {
    data_t * found;
    add_record("Waade is awesome",1,0,0,0,0);
    add_record("Taylor Swift", 2,0,0,0,0);
    add_record("Freedom from the Known",3,0,0,0,0);
    add_record("Panchtantra",4,0,0,0,0);
    while(1)
    {
        LCD_Menu();
        char x=keyboard();
        LCD_Init();
```

```
        LCD_String("User ID");
        char y=keyboard();
        for(int i=0;i<=9000;i++)
        asm("NOP");
        LCD_Init();
        for(int i=0;i<=9000;i++)
        asm("NOP");
        LCD_String("Book ID");
        char z=keyboard();
        for(int i=0;i<=9000;i++)
        asm("NOP");
        int u_id=(int) y;
        int b_id=(int) z;
        switch(x)
        {

            case '1': borrow(b_id,u_id);
            break;
            case '2': return_book(b_id);
            break;
            default:  LCD_Init();
            LCD_String("Error");
            break;
        }
    UART_init(9600);    // initialise UART communication

    UART_sendString(itoa(fine,str,10));



    }
    return 0;

}
```

**WIFI MODULE CODE: FOR SENDING AN EMAIL USING SMTP**

```
#include <ESP8266WiFi.h>
const char* ssid = "A1";          // Enter the SSID of your WiFi Network.
******************
const char* password = "12345678";     // Enter the Password of your WiFi
Network. ********************
char server[] = "mail.smtp2go.com";     // write SMTP server address
******************
byte sendEmail();
byte emailResp();
WiFiClient Client;               //define wifi client as client
String variable = "";
void setup() {
  delay(1000);
  Serial.begin(9600);          // set baudrate as 9600 similar to
Atmega16
  Serial.println("");
  Serial.print("Connecting To: ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);     // connect to WIFI

  while (WiFi.status() != WL_CONNECTED)
  {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi Connected.");
  Serial.print("IP address: ");
  Serial.println(WiFi.localIP());

}


void loop() {
  char inChar = 0;
  char inData[6] = "";        // data length of 6 characters

  String variable1 = "";
  int index1 = 0;

  if ( Serial.available() > 0 ) {                      // Read from Rx
from atmega16
    while (Serial.available() > 0 && index1 < 6)     // read till 6th
character
    {
      delay(100);
      inChar = Serial.read();      // start reading serilly and save to
variable
      inData[index1] = inChar;
      index1++;
```

```
      inData[index1] = '\0';         // Add a null at the end
    }
    variable.toUpperCase();       // convert to uppercase
    for (byte  i = 0 ; i < 6 ; i++) {
      variable.concat(String(inData[i]));    // concat strings
    }
    Serial.print("Fine = "); Serial.println(variable);  // debug and
print incoming data
    delay(20);
  }
  String  string = String(variable);                          //
string used to compare
  sendEmail();                                // send email by calling
function
  Serial.print("Mail sent to:"); Serial.println(" The recipient");
// debug if sent
  Serial.println("");

  }



byte sendEmail()
{
  if (Client.connect(server, 2525) == 1)        // connect to smtp server
with port address 2525
  {
    Serial.println(F("connected to server"));
  }
  else
  {
    Serial.println(F("connection failed"));
    return 0;
  }
  if (!emailResp())         // if connection failed return now
    return 0;
  //
  Serial.println(F("Sending EHLO"));
  Client.println("EHLO www.example.com");      // Send command EHLO
previosly it was HELO*******************
  if (!emailResp())
    return 0;

  Serial.println(F("Sending auth login"));
  Client.println("AUTH LOGIN");
  if (!emailResp())
    return 0;
  //
  Serial.println(F("Sending User"));
  // Change this to your base64, ASCII encoded SMTP username.
  //For example, the email address dummy@gmail.com encoded as
djfdBDBEDEJD545616vfbSJHB=

  Client.println("a3NoaXByYS5wcmFzYWRoQGdtYWlsLmNvbQ=="); //base64,
ASCII encoded SMTP Username    *******************
  if (!emailResp())
```

```
    return 0;

  Serial.println(F("Sending Password"));
  // change to your base64, ASCII encoded SMTP password
 // For example, the the password "password" encoded as
IBjbjHUInOUi4654==

  Client.println("Tml0aHlhMTIzNDE=");   //base64, ASCII encoded SMTP
Password               ********************

  if (!emailResp())
    return 0;
  //
  Serial.println(F("Sending From"));
  // change to sender email address
  Client.println(F("MAIL From: kshipra.prasadh@gmail.com"));       //
********************
  if (!emailResp())
    return 0;
  // change to recipient address
  Serial.println(F("Sending To"));
  Client.println(F("RCPT To: c.maouli2017@gmail.com"));    //
********************

  if (!emailResp())
    return 0;
  //
  Serial.println(F("Sending DATA"));
  Client.println(F("DATA"));
  if (!emailResp())
    return 0;
  Serial.println(F("Sending email"));
  // change to recipient address
  Client.println(F("To:  c.maouli2017@gmail.com"));                //
********************
  // change to your address
  Client.println(F("From: kshipra.prasadh@gmail.com"));
//********************
  Client.println(F("Subject: ESP8266 test e-mail\r\n"));
  if(variable=="0")
  {Client.println(F("No Fine amount"));}
  else
  {Client.println(F("Please pay the fine"));
  //
  Client.println(F("."));
  if (!emailResp())
    return 0;
  //
  Serial.println(F("Sending QUIT"));
  Client.println(F("QUIT"));
  if (!emailResp())
    return 0;
  //
  Client.stop();
  Serial.println(F("disconnected"));
  return 1;
```

```
}
}

byte emailResp()
{
  byte responseCode;
  byte readByte;
  int loopCount = 0;

  while (!Client.available())
  {
    delay(1);
    loopCount++;
    // Wait for 20 seconds and if nothing is received, stop.
    if (loopCount > 20000)
    {
      Client.stop();
      Serial.println(F("\r\nTimeout"));
      return 0;
    }
  }

  responseCode = Client.peek();
  while (Client.available())
  {
    readByte = Client.read();
    Serial.write(readByte);
  }

  if (responseCode >= '4')
  {
    //  efail();
    return 0;
  }
  return 1;
}
```

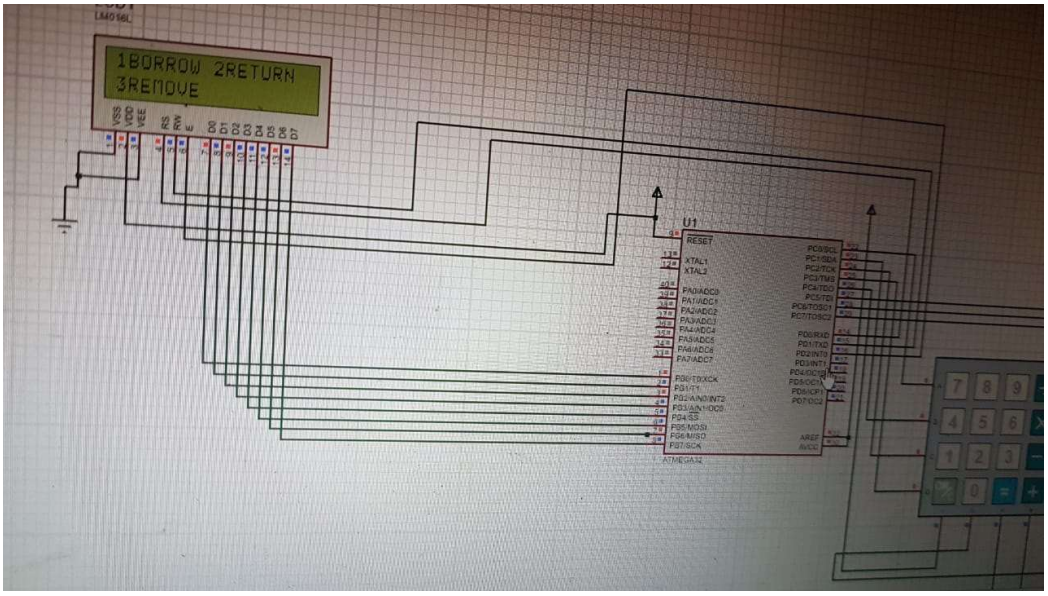
# CHALLENGES FACED

- The hardware of the project was unable to work
- The reason could be the defective ATMEGA chip.
- The LCD, Keypad and other components were individually tested.
- Even the microcontroller responded to signals in individual ports, however, on more research, we got to know that certain individual pins of the microcontroller can get damaged, leading to non functionality.

TESTS DONE

1. All the LCD PORTS pins were tested.
2. The ESP 8266 module was tested independently- it worked fine
3. PORTS A,B, C and D of the AVR were tested and showed no problem
4. The code burnt into the ATMEGA, through ATMEL studio. The ATMEL studio indicated that the burn was successful.
5. All libraries were checked for compatibility with the hardware: They are compatible
6. The code runs and builds successfully.
7. The Proteus simulation works perfectly

# RESULTS




In PROTEUS, all functionalities, including borrow, return and fines are verified.

**REFERENCES**


1. Ninth International Conference on Complex, Intelligent, and Software Intensive Systems 978-1-4799-8870-9/15 $31.00 © 2015 IEEE DOI 10.1109/CISIS.2015.15 105 2 An RFID-based System for Library Management and Its Performance Evaluation Kiyotaka FUJISAKI Dept. of Advanced Information Technology Kyushu University Fukuoka, Japan Email: fujisaki@ait.kyushu-u.ac.jp
2. Proceedings of 2013 IEEE Conference on Information and Communication Technologies (ICT 2013),.

3. An Internet Based RFID Library Management System A.Pravin Renold1 , Joshi Rani.R2 Department of TIFAC-CORE in Pervasive Computing Technologies, Velammal Engineering College, Anna University Chennai, Tamil Nadu, India. 1 pravinrenold.tifac@velammal.edu.in 2 joshirani.r@gmail.com [3]

4. Intelligent Book Positioning For Library Using RFID And Book Spine Matching WING W. Y. NG, YI-SONG QIAO, LI LIN, HAI-LAN DING, PATRICK P. K. CHAN, DANIEL S. YEUNG Machine Learning and Cybernetic Research Center, School of Computer Science and Engineering, South China University of Technology, Guangzhou, China E-MAIL: wingng@ieee.org, yis.qiao@ieee.org

5. Mobility Improves Performance of RFID Library Systems Konstantinos Tountas1 , Antonis G. Dimitriou2 , Aggelos Bletsas1 and John N. Sahalos4 1School of ECE, Technical Univ. of Crete, Greece, ktountas@isc.tuc.gr, aggelos@telecom.tuc.gr 2Dept. of ECE, Aristotle Univ. of Thessaloniki, Greece,

6. Sameera.S, Saranya.S, and Vimalraj.S, 'RFID based Library Management System' from International Journal of Innovative Research in Advanced Engineering (IJIRAE) ISSN: 2349-216 Issue 1, Volume 2 (January 2015).

7. Ashim A Patil, 'i-TEK RFID Based Library Management Suite' a White paper. byInfotek Software & Systems PLtd, Pune, India (February, 2017)

8. **AVR Microcontrollers, Mazidi and Mazidi**