



e-ISSN: 2319-8753 | p-ISSN: 2320-6710

IJRSET

International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY


INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 9, Issue 9, September 2020

ISSN INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA

Impact Factor: 7.512

 9940 572 462

 6381 907 438

 ijirset@gmail.com

 www.ijirset.com

Analysis of TDM for Analog Signal with the help of CMOS IC4051

Shruti Panchal¹, P.D.Lele²

M. Phil Student, Department of Physics, Gujarat University, Ahmedabad, Gujarat, India¹

Associate Professor, Department of Physics, Gujarat University, Ahmedabad, Gujarat, India²

ABSTRACT: Multiple signals are transmitted simultaneously using Time Division Multiplexing (TDM). TDM circuit can be constructed using CMOS IC 4051. This circuit have mainly two parts, one is transmitter and another one is receiver end. We can use different type of input signals in the transmitter part and the control block for manage gain. This different input signal we can simultaneously apply in the multiplexer block. The output of the multiplexer block becomes the input of the de-multiplexer block. Then again we use control block at the receiver end for controlling the gain and get output. We are going through basic experiment to advance experiment. Here we also construct basic to advance level application using multiplexer.

KEYWORDS: TDM, CMOS, Multiplexer, De-multiplexer, control block.

I. INTRODUCTION

A multiplexing technique by which multiple data signals can be transmitted over a common communication channel in different time slots is known as Time Division Multiplexing (TDM). Because of TDM system we save time, reduce circuit complexity, reduce the cost and most important thing is we can perform number of operation at a time. TDM system is fundamental building block of many complex circuitries. There are mainly two parts in TDM system: 1. Transmitter and 2. Receiver. TDM systems are work for analog as well as digital signal. But the TDM circuit designs are quite different in both cases analog & digital. TDM system for analog signal consist two block transmitter and receiver. Transmitter consist input signal generator, clock circuit, multiplexer and the control blocks. Similarly receiver block consist de-multiplexer and control block. TDM system for digital signal is almost same as analog system but at the receiver end we only get information in term of ones and zeros, because we use logic gates as the basic building blocks of the circuit diagram.

II. RELATED WORK

Time division multiplexing is a useful technique, which has found many application in the telecommunication area.[1] In 2010, Gilbert Hendry et al suggested Time Division Multiplexing for overcome heavy loads and an overhead in power dissipation.[2] In 1998 Satoki Kawanishi said that any optical time division multiplexed (OTDM) including some part which is time division multiplexing, time division de-multiplexing, signal generation, timing extraction, and waveform measurement.[3] In 2005, Laung-Terng Wang describe that time division de-multiplexing and multiplexing of high-data-rate scan patterns applied on I/O's into low-data-rate scan patterns applied on Virtual Scan compression circuitry to further reduce test application time and test pin-count without coverage loss.[4] In 1990, S. Aikoh & A. Kanemasa derived a method, which is Time Division Multiplexing method with channel compression and a device therefor. They explain that how this method is working. They take N-channel sample at M-time slot, but the requirement is M is equal to or less than N. [5] In 1991, Y. Kawamura has given one more application of TDM, which is FDM-TDM transforming device. [6] A. B. Wiczorek et al. has been given one application of TDM, which is energy saving protocol for a TDM Radio. [7] In 1994, Kabaya et al. describe stuff-synchronization TDM system. In the stuff synchronization TDM system they suggested an incoming TDM signal is delayed so that a plurality of TDM frame sequences are successively delayed by one time slot and appear at the inputs of a data selector. A control circuit is provided for deriving a channel-by-channel selection signal from stuff specification bits contained in the incoming TDM signal. The data selector is responsive to the selection signal to select a channel in one of the TDM frame sequences and delivers the selected channel as a TDM output signal to a time division switch, or cross-connect in which it is decomposed into channels for switching to a desired route. [8] In 2004, Suzuki et al., explain TDM multiplexing & de-multiplexing and how to convert TDM/WDM & WDM/TDM.[9] In 2007, G. Katz and D. Sadot Proposed a method which is time division multilevel multiplexing communication method. In this method they describe

that how we can increase channel capacity. For that we have to combine time division with multilevel signals and this way we get better channel capacity. This method also increases bandwidth of the signal/channel. [10]

The work in this paper is divided in to two parts: 1) Design PCB using proteas software 2)Fundamental circuit blocks 3) application of this circuit.Fundamental circuit blocks includes wein bridge oscillator, astable multivibrator, gain control, multiplexer, de-multiplexer and amplifier circuit. We can calculate individual circuit parameter also. In the next part the construction of application using multiplexing concept. In this part we move very basic to advance level of multiplexing.

III. METHODOLOGY

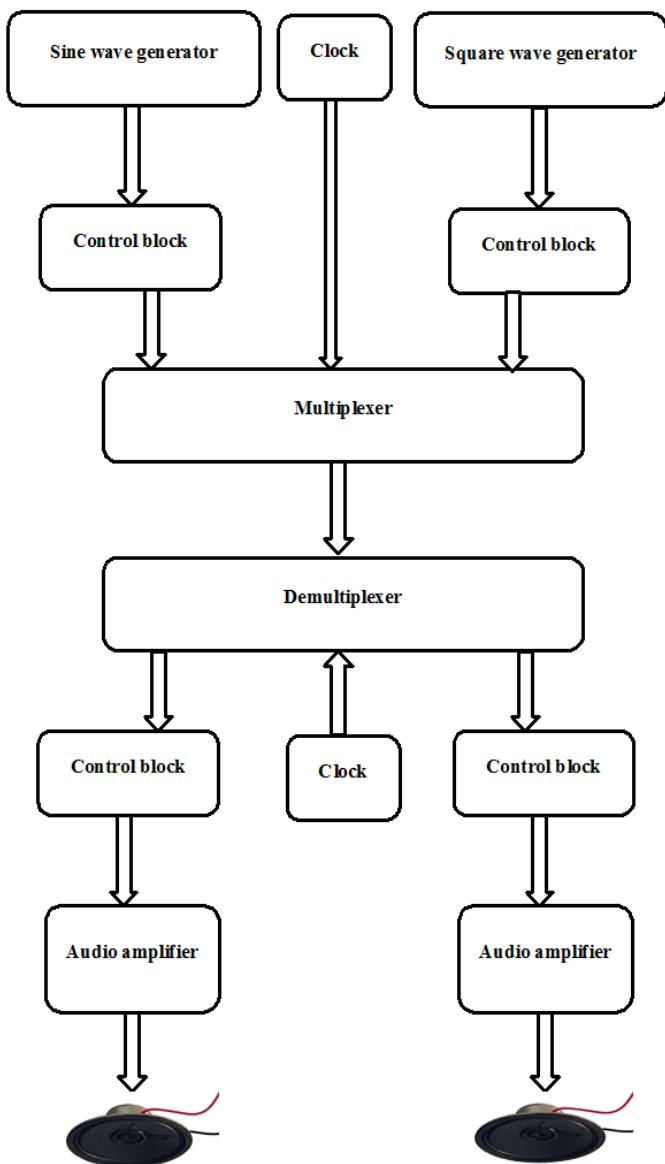


Figure 1 : Block diagram of system

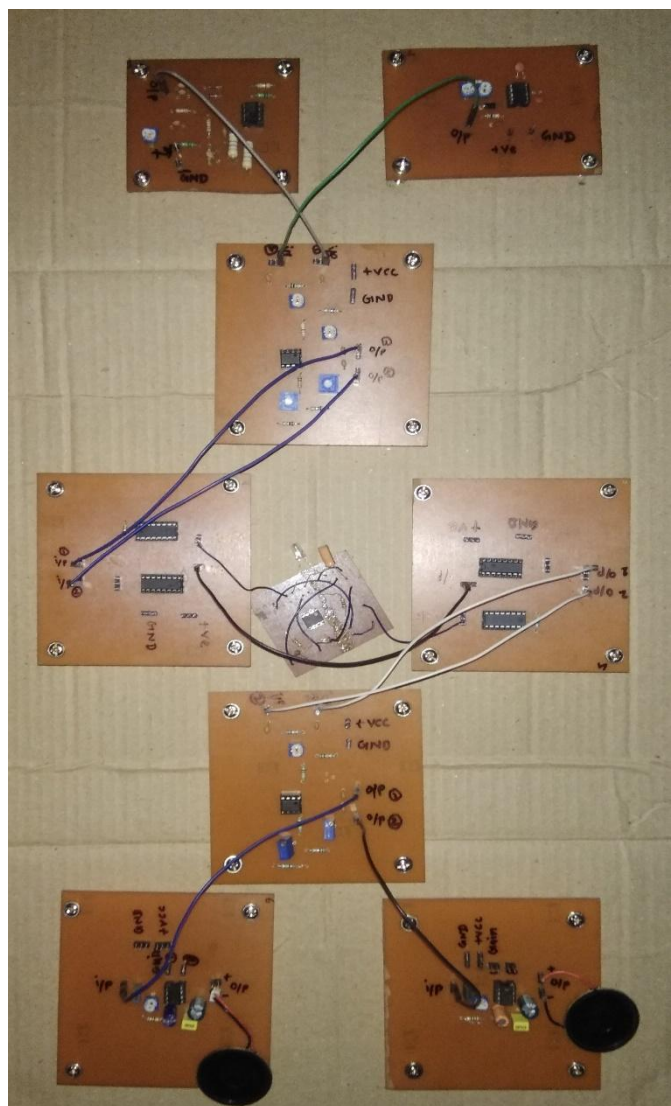


Figure 2: Circuit diagram of system

As shown in the block diagram many senders send the signal for a particular time, there individual information are resorted at the receiver part and then distributed to the particular receiver. Where the sender and receiver have a different multiple positions switches which can control the signal flow path. At the transmitter/sending end, the switch

takes signal of the user and put it in the link/cable. And at the receiver end the switch can takes the signal from the link and distributes to the particular point. This way we can share too many signals at a time and each users are independent to the others.

In this experiment, we use the TDM switch for the analog communication system. We need clock circuit, counter circuit using IC4029 and a switch to perform this experiment. As shown in the block diagram, sine wave and square wave are the two input of the multiplexer. We use IC 4051 for the multiplexer purpose, which is 8 - channel IC. In IC 4051 pin number 9, 10 & 11 are select pin. Pin number 3 is output pin and pin number 1, 2, 4, 5, 12, 13, 14 & 15 are the 8 input pin. Out of this eight channel we only use two channel here. Multiplexer is important block of this experiment. The counter 4029 is drive through clock pulse from IC555. The output of the counter circuit is given to the pin number 10 and pin number 11 of the IC 4051. This counter produces the counts from 000 to 111. This counts are connected with the multiplexer for the decoding. The switch will allow only one user signal from the many users with each count and repeat this in sequential manner. The counter turns the count and we get one by one different position. That's why we get only one signal at a time. At the receiver end we use de-multiplexer switch, which is similar to the multiplexer configuration also for the de-multiplexer purpose we use IC 4051. In IC 4051 pin number 9, 10 & 11 are select pin. Pin number 3 is input pin and pin number 1, 2, 4, 5, 12, 13, 14 & 15 are the 8 output pin. We get the different output signals at the output pins from IC 4051. Here we use only two pin number 13 & pin number 15 out of 8 - channel.

Application:-1 Smart Alarm with person counter using Switch (IC 4051).

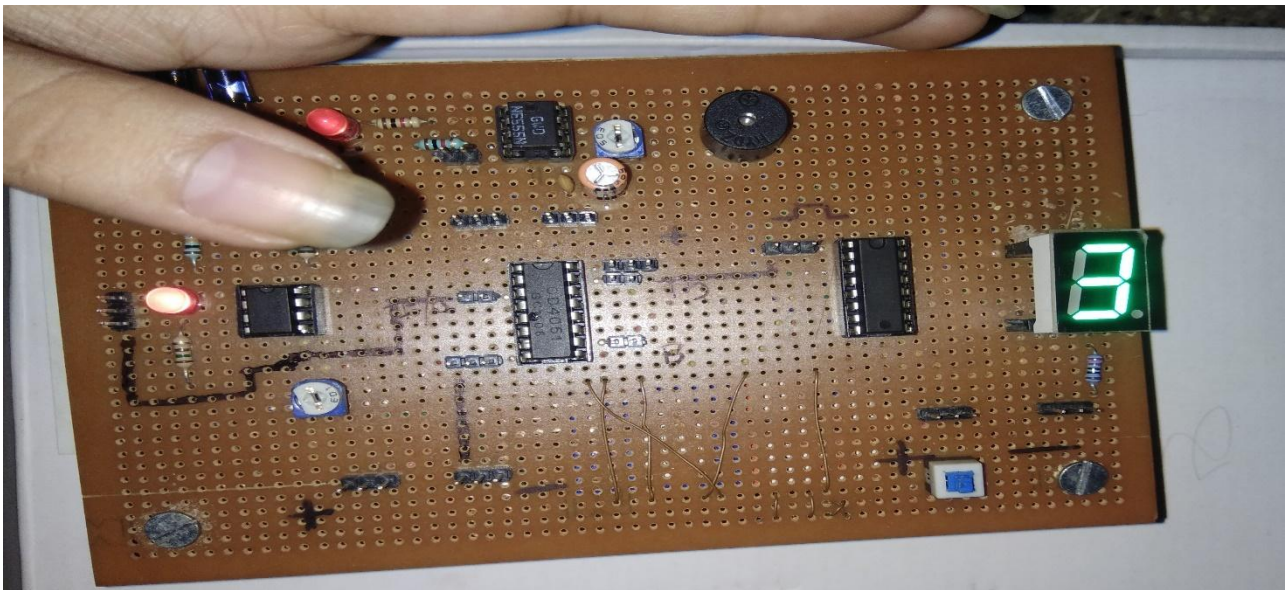


Figure: 3 - Smart Alarm

There are five block in this application.1) Detector circuit 2) IC 555 as Monostable 3)IC 4051 as Switch 4) IC 4026 Decade Counter 5)Seven Segment.In detector circuit, there are two parts in circuit, one is transmitter and another is receiver part include IR module. This IR module transmits continuously IR rays and at the receiver end include IR module. This IR receiver module receives IR rays. This output can be applied to comparator circuit. Here we use LM 358 IC as comparator circuit. When receiver does not receive any signal then variable resistor at the inverting terminal of IC LM 358 goes higher than non-inverting terminal of that IC. The output of comparator goes low and LED does not glow. When receiver receive a signal then variable resistor at the inverting input goes lower than non-inverting terminal, thus output of the comparator goes higher and LED start to glowing. The resistor R1, R2 & R3 is used to maintain 10 mA current in photodiode and LEDs. Variable resistor VR1 is used to set the sensitivity in circuit diagram and VR2 is used to adjust the output.In this circuit diagram we use 4026 decade counter, this counter is receives a clock pulse signal from timer IC 555. We use switch circuit using 4051 multiplexer circuit. Here 555 timer IC connect in monostable state, which provide each time clock signal to decade counter. We take output from pin number 3 of monostable circuit and apply as an input to 4026 decade counter. This counter is connected with seven segment display, this display provide person number count.

Application:-2 Water level indicator using IC 4051.

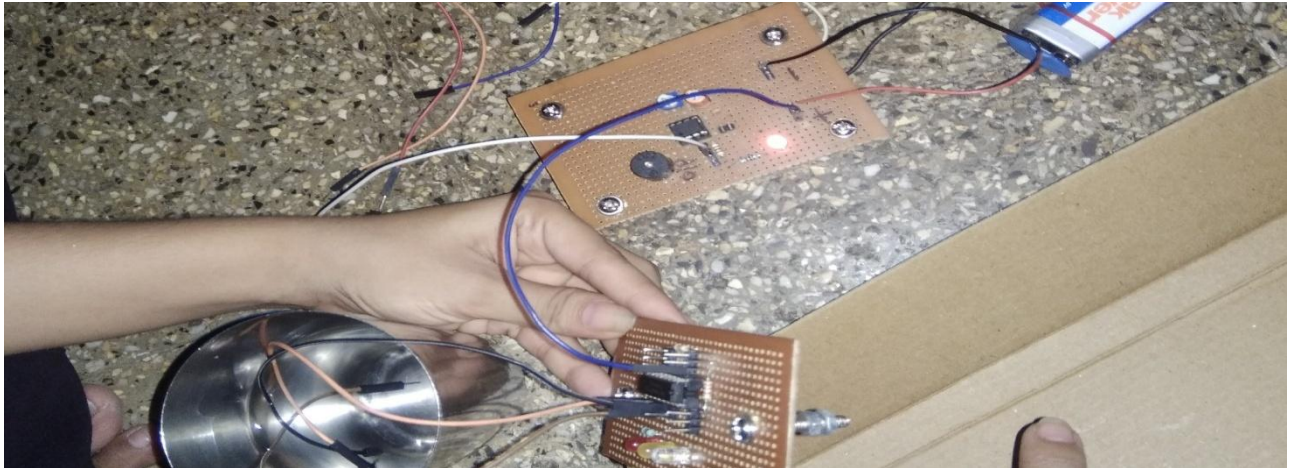


Figure: 4 - Water level indicator

There are two block in this application. 1) IC 555 as Monostable 2) IC 4051 as switch. In this application we use three pin number 9, 10, 11 as a select line of multiplexer IC4051. We can apply input on pin number 3 of IC 4051. Pin number 8 & 16 of IC 4051 is ground and positive respectively. Here we use low level logic, for this we use voltage divider circuit between pin number 3 and ground, to divide 5 V in to 2.5 V. We can use two wire as a switch, one from +V_{CC} & another from select pin number of IC 4051. This wire work as switch, if water level is down circuit will open & water level up circuit will short. Whenever these wire become short, the output become high. In the output we use 3 LEDs as Green on pin number 15, yellow on pin number 13 and red on pin number 4 of the IC 4051. Also we use buzzer for the indication over here, this buzzer is connected with monostable block, which provide particular time period for the buzzer ring.

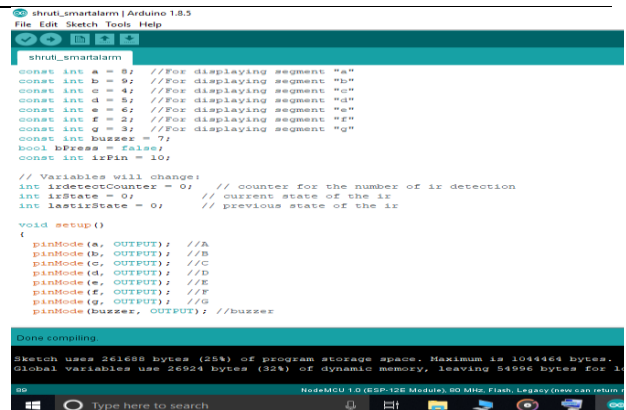
IV. EXPERIMENTAL RESULTS& CONCLUSION

We have performed TDM experiments for analog signal. In this system we get waveform at output. In the first two circuits we get sine wave & square wave respectively. Then control block maintain gain/amplitude and apply to the input of multiplexer circuit. At the output of the multiplexer circuit we get sine wave as well as square wave in particular time period. This output is apply to the input of the de-multiplexer circuit. The de-multiplexer circuit give individual signal to the particular channel. Then if we want to amplify the signal apply to the amplifier block. We can calculate many parameters from this circuit & also plot the graph. I will calculate this parameter in next paper of TDM for analog signals. In applications we get output from basic circuitry as well as from advance programming also.

Programme for application -1:

```
const int a = 8; //For displaying segment "a"
const int b = 9; //For displaying segment "b"
const int c = 4; //For displaying segment "c"
const int d = 5; //For displaying segment "d"
const int e = 6; //For displaying segment "e"
const int f = 2; //For displaying segment "f"
const int g = 3; //For displaying segment "g"
const int buzzer = 7;
bool bPress = false;
const int irPin = 10;

// Variables will change:
int irdetectCounter = 0; // counter for the number of ir
```





```

detection
int irState = 0;    // current state of the ir
int lastirState = 0; // previous state of the ir

void setup()
{
  pinMode(a, OUTPUT); //A
  pinMode(b, OUTPUT); //B
  pinMode(c, OUTPUT); //C
  pinMode(d, OUTPUT); //D
  pinMode(e, OUTPUT); //E
  pinMode(f, OUTPUT); //F
  pinMode(g, OUTPUT); //G
  pinMode(buzzer, OUTPUT); //buzzer

  pinMode( irPin , INPUT_PULLUP );
  Serial.begin(9600);
  displayDigit(irdetectCounter);
}

void loop()
{
  irState = digitalRead(irPin);
  // compare the irState to its previous state
  if (irState != lastirState) {
    // if the state has changed, increment the counter
    if (irState == LOW)
    {
      // if the current state is HIGH then the ir went from off to
on:
      bPress = true;
      irdetectCounter++;
      if( irdetectCounter > 9) irdetectCounter =0 ;
      digitalWrite(buzzer,HIGH);
      delay(10000);          // wait for a 10 second
      digitalWrite(buzzer,LOW);
      Serial.println("on");
    }
    else
    {
      // if the current state is LOW then the ir went from on to off:
      Serial.println("off");
    }
    // Delay a little bit to avoid bouncing
    delay(50);
  }
}
    
```

shrut_i_smartalarm | Arduino 1.8.5

```

File Edit Sketch Tools Help

shrut_i_smartalarm

pinMode(buzzer, OUTPUT); //buzzer

pinMode( irPin , INPUT_PULLUP );
Serial.begin(9600);
displayDigit(irdetectCounter);
}

void loop()
{
  irState = digitalRead(irPin);
  // compare the irState to its previous state
  if (irState != lastirState) {
    // if the state has changed, increment the counter
    if (irState == LOW)
    {
      // if the current state is HIGH then the ir went from off to on:
      bPress = true;
      irdetectCounter++;
      if( irdetectCounter > 9) irdetectCounter =0 ;
      digitalWrite(buzzer,HIGH);
      delay(10000);          // wait for a 10 second
      digitalWrite(buzzer,LOW);
      Serial.println("on");
    }
    else
    {
      // if the current state is LOW then the ir went from on to off:
    }
  }
}
    
```

Done compiling.

Sketch uses 261688 bytes (25%) of program storage space. Maximum is 1044464 bytes. Global variables use 26924 bytes (32%) of dynamic memory, leaving 54996 bytes for 1

89 NodeMCU 1.0 (ESP-12E Module), 80 MHz, Flash, Legacy (new can return r

shrut_i_smartalarm | Arduino 1.8.5

```

File Edit Sketch Tools Help

shrut_i_smartalarm

    // if the current state is LOW then the ir went from on to off:
    Serial.println("off");
  }
  // Delay a little bit to avoid bouncing
  delay(50);
}
// save the current state as the last state, for next time through the loop
lastirState = irState;

if( bPress ){
  turnOff();
  displayDigit(irdetectCounter);
}
}

void displayDigit(int digit)
{
  //Conditions for displaying segment a
  if(digit!=1 && digit != 4)
  digitalWrite(a,HIGH);

  //Conditions for displaying segment b
  if(digit != 5 && digit != 6)
  digitalWrite(b,HIGH);

  //Conditions for displaying segment c
}
    
```

Done compiling.

Sketch uses 261688 bytes (25%) of program storage space. Maximum is 1044464 bytes. Global variables use 26924 bytes (32%) of dynamic memory, leaving 54996 bytes for 1

89 NodeMCU 1.0 (ESP-12E Module), 80 MHz, Flash, Legacy (new can return r



```

}
// save the current state as the last state, for next time through
the loop
lastirState = irState;

if( bPress ){
    turnOff();
    displayDigit(irdetectCounter);
}
}

void displayDigit(int digit)
{
//Conditions for displaying segment a
if(digit!=1 && digit != 4)
digitalWrite(a,HIGH);

//Conditions for displaying segment b
if(digit != 5 && digit != 6)
digitalWrite(b,HIGH);

//Conditions for displaying segment c
if(digit !=2)
digitalWrite(c,HIGH);

//Conditions for displaying segment d
if(digit != 1 && digit !=4 && digit !=7)
digitalWrite(d,HIGH);

//Conditions for displaying segment e
if(digit == 2 || digit ==6 || digit == 8 || digit==0)
digitalWrite(e,HIGH);

//Conditions for displaying segment f
if(digit != 1 && digit !=2 && digit!=3 && digit !=7)
digitalWrite(f,HIGH);

//Conditions for displaying segment g
if (digit!=0 && digit!=1 && digit !=7)
digitalWrite(g,HIGH);
}
void turnOff()
{
digitalWrite(a,LOW);
digitalWrite(b,LOW);
digitalWrite(c,LOW);
digitalWrite(d,LOW);
digitalWrite(e,LOW);
digitalWrite(f,LOW);
digitalWrite(g,LOW);
}
}
    
```

```

shruti_smartalarm | Arduino 1.8.5
File Edit Sketch Tools Help

shruti_smartalarm
//Conditions for displaying segment b
if(digit != 5 && digit != 6)
digitalWrite(b,HIGH);

//Conditions for displaying segment c
if(digit !=2)
digitalWrite(c,HIGH);

//Conditions for displaying segment d
if(digit != 1 && digit !=4 && digit !=7)
digitalWrite(d,HIGH);

//Conditions for displaying segment e
if(digit == 2 || digit ==6 || digit == 8 || digit==0)
digitalWrite(e,HIGH);

//Conditions for displaying segment f
if(digit != 1 && digit !=2 && digit!=3 && digit !=7)
digitalWrite(f,HIGH);

//Conditions for displaying segment g
if (digit!=0 && digit!=1 && digit !=7)
digitalWrite(g,HIGH);
}
void turnOff()
{
digitalWrite(a,LOW);
}

Done compiling.
Sketch uses 261688 bytes (25%) of program storage space. Maximum is 1044464 bytes.
Global variables use 26924 bytes (32%) of dynamic memory, leaving 54996 bytes for 1
89 NodeMCU 1.0 (ESP-12E Module), 80 MHz, Flash, Legacy (new can return
    
```

```

shruti_smartalarm | Arduino 1.8.5
File Edit Sketch Tools Help

shruti_smartalarm
//Conditions for displaying segment d
if(digit != 1 && digit !=4 && digit !=7)
digitalWrite(d,HIGH);

//Conditions for displaying segment e
if(digit == 2 || digit ==6 || digit == 8 || digit==0)
digitalWrite(e,HIGH);

//Conditions for displaying segment f
if(digit != 1 && digit !=2 && digit!=3 && digit !=7)
digitalWrite(f,HIGH);

//Conditions for displaying segment g
if (digit!=0 && digit!=1 && digit !=7)
digitalWrite(g,HIGH);
}
void turnOff()
{
digitalWrite(a,LOW);
digitalWrite(b,LOW);
digitalWrite(c,LOW);
digitalWrite(d,LOW);
digitalWrite(e,LOW);
digitalWrite(f,LOW);
digitalWrite(g,LOW);
}

Done compiling.
Sketch uses 261688 bytes (25%) of program storage space. Maximum is 1044464 bytes.
Global variables use 26924 bytes (32%) of dynamic memory, leaving 54996 bytes for 1
89 NodeMCU 1.0 (ESP-12E Module), 80 MHz, Flash, Legacy (new can return
    
```



Programme of application – 2:

<pre> const int buzzer = 6; const int Switch1 = 2; const int Switch2 = 3; const int Switch3 = 4; // the number of the pushbutton pin const int green = 13; const int yellow = 12; const int red = 11; // the number of the LED pin // variables will change: int buttonState = 0; int buttonState1 = 0; int buttonState2 = 0; // variable for reading the pushbutton status void setup() { // initialize the LED pin as an output: pinMode(green, OUTPUT); pinMode(yellow, OUTPUT); pinMode(red, OUTPUT); digitalWrite(buzzer, LOW); // initialize the pushbutton pin as an input: pinMode(Switch1, INPUT); pinMode(Switch2, INPUT); pinMode(Switch3, INPUT); } void loop() { buttonState = digitalRead(Switch1); buttonState1 = digitalRead(Switch2); buttonState2 = digitalRead(Switch3); if (buttonState == HIGH) { // turn LED on: digitalWrite(green, HIGH); } else { // turn LED off: digitalWrite(green, LOW); } if (buttonState == HIGH && buttonState1 == HIGH) { digitalWrite(yellow, HIGH); } else { // turn LED off: digitalWrite(yellow, LOW); } if (buttonState == HIGH && buttonState1 == HIGH && buttonState2 == HIGH) { digitalWrite(red, HIGH); digitalWrite(buzzer, HIGH); } else { // turn LED off: digitalWrite(red, LOW); digitalWrite(buzzer, LOW); } } </pre>	 <p>The screenshot shows the Arduino IDE interface with the code from the left column. The 'Done compiling' message indicates successful compilation. The memory usage summary shows: Sketch uses 256356 bytes (24%) of program storage space. Maximum is 1044464 bytes. Global variables use 26796 bytes (32%) of dynamic memory, leaving 55124 bytes for local variables.</p>
<pre> // turn LED on: digitalWrite(green, HIGH); } else { // turn LED off: digitalWrite(green, LOW); } if (buttonState == HIGH && buttonState1 == HIGH) { digitalWrite(yellow, HIGH); } else { // turn LED off: digitalWrite(yellow, LOW); } if (buttonState == HIGH && buttonState1 == HIGH && buttonState2 == HIGH) { digitalWrite(red, HIGH); digitalWrite(buzzer, HIGH); } else { // turn LED off: digitalWrite(red, LOW); digitalWrite(buzzer, LOW); } } </pre>	 <p>The screenshot shows the Arduino IDE interface with the code from the left column. The 'Done compiling' message indicates successful compilation. The memory usage summary shows: Sketch uses 256356 bytes (24%) of program storage space. Maximum is 1044464 bytes. Global variables use 26796 bytes (32%) of dynamic memory, leaving 55124 bytes for local variables. Maximum is 91520 bytes.</p>
<pre> // turn LED off: digitalWrite(red, LOW); digitalWrite(buzzer, LOW); } } </pre>	 <p>The screenshot shows the Arduino IDE interface with the code from the left column. The 'Done compiling' message indicates successful compilation. The memory usage summary shows: Sketch uses 256356 bytes (24%) of program storage space. Maximum is 1044464 bytes. Global variables use 26796 bytes (32%) of dynamic memory, leaving 55124 bytes for local variables. Maximum is 91520 bytes.</p>



ACKNOWLEDGMENT

The author is thankful to department of science and technology (DST) and DRS-SAP for providing experimental facilities. Author is thankful to Prof. P. N. GAJJAR head of the physics department, school of sciences, Gujarat University, Ahmedabad for his constant encouragement. Author is also thankful to Prof.P.D.LELE for providing continuous guidance and knowledge.

REFERENCES

- [1] H. Stark, F.B Tuteur, J. B. Anderson, "Modern Electrical Communications", 2nd Ed. New Jersey: Prentice Hall, 1988.
- [2] G. Hendry et al, "Silicon Nano photonic Network-On-Chip Using TDM Arbitration", 18th IEEE Symposium on High Performance Interconnects, 2010.
- [3] S. Kawanishi, "Ultrahigh-Speed Optical Time Division Multiplexed Transmission Technology Based on Optical Signal Processing", IEEE Journal of Quantum Electronics, Vol. 34, NO. 11, NOVEMBER 1998.
- [4] Laung-Terng Wang et al, "Ultra Scan: Using Time-Division Demultiplexing/Multiplexing (TDDM/TDM) with Virtual Scan for Test Cost Reduction", IEEE International Test Conference, 2005
- [5] S. Aikoh & A. Kanemasa, "Time Division Multiplexing Method with Channel Compression and a Device Therefor", U.S Patent, Jun.12, 1990.
- [6] Y. Kawamura, "FDM-TDM Transforming Device", U.S Patent, Aug. 6, 1991.
- [7] A. B. Wiczorek et al, "Energy Saving Protocol for a TDM Radio", U.S Patent, Sep.22, 1992.
- [8] E. Kabaya & K. Kurokawa, "Phase Alignment Circuit for Stuffed - Synchronized TDM Transmission System with Cross - Connect Function", U. S. Patent, Apr.19, 1994.
- [9] M. Suzuki & N. Edagawa, "Optical TDM Multiplexer, Optical TDM Demultiplexer, WDM/TDM Converter and TDM/WDM Converter", U. S. Patent, Aug.10, 2004
- [10] G. Katz & D. Sadot, "Time-Division Multilevel Multiplexing Communication Method", IEEE Photonics Technology Letters, Vol. 19, NO. 20, Oct. 15, 2007.
- [11] P. D. Lele, "Some Experiments in Communication using Fiber Optics", Physics Education, Oct. 2001 – March 2002.
- [12] https://create.arduino.cc/projecthub/arduino_uno_cool/create-a-7-segment-display-254ebe



INNO  **SPACE**
SJIF Scientific Journal Impact Factor

Impact Factor:
7.512

ISSN INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 **9940 572 462**  **6381 907 438**  **ijirset@gmail.com**



www.ijirset.com

Scan to save the contact details