

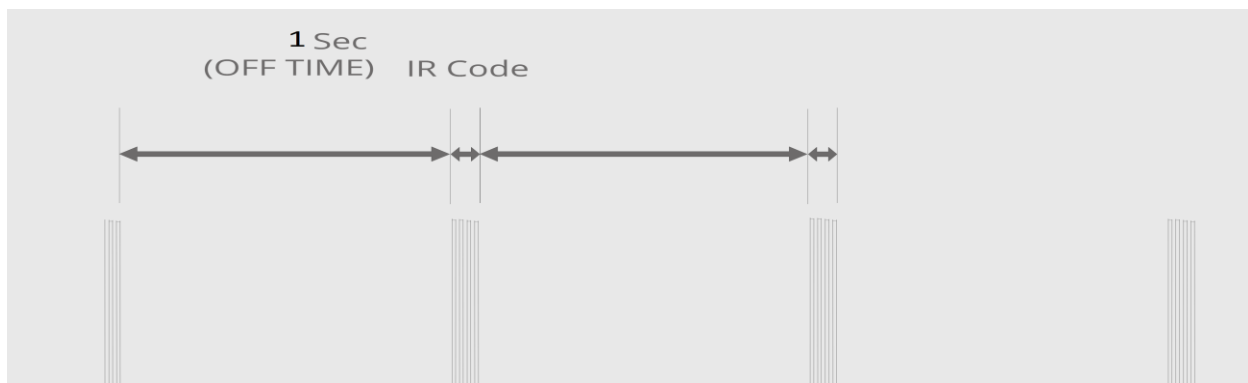
IR Code Decoding

The actual code to decode your signals

Data sequencing and Timing diagram:-

At every required node there will an IR LED which will continuously emit IR codes at frequency of 1 Hz i.e. a burst of IR signal will be send through the transmitter at every 1 second which will contain the data to the next food location relative to the current node location.

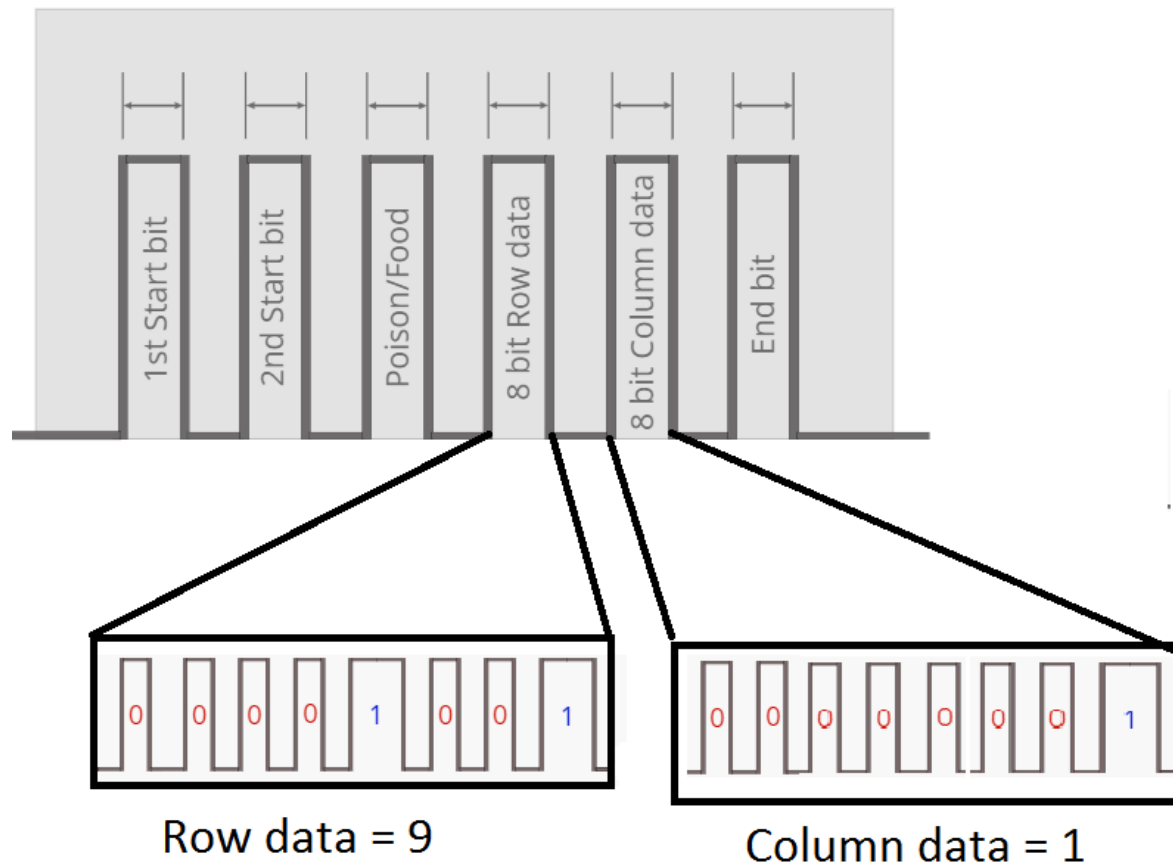
Each burst of IR signal consists of a 20 bit data sequence which comprises of two START bits, one FOOD/POISON bit, eight bits of ROW data, and eight bits of COLUMN data plus at last a stop bit.



Breakdown of the IR Code

The sequence of the IR signal will be as following:-

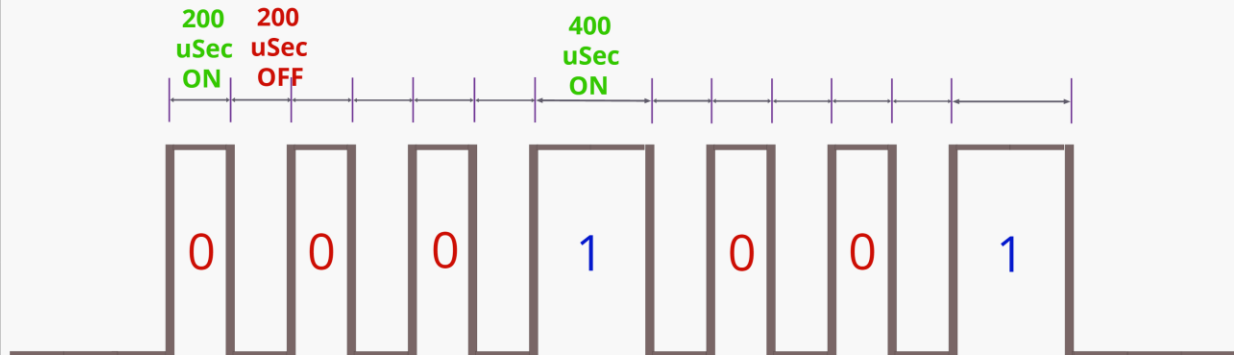
<Start bit 1> <Start bit 1> {Row Data} {Column Data} <Stop Bit>



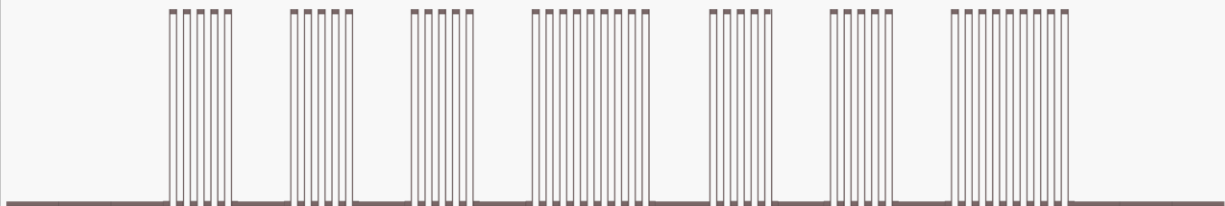
The code sent by the transmitter will vary in length. Each low time and high time is **200uS** long.

General Representation of Data

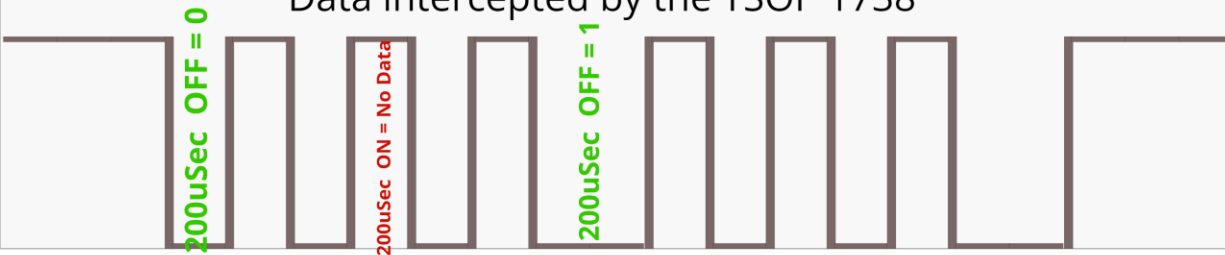
Data that has to be transmitted



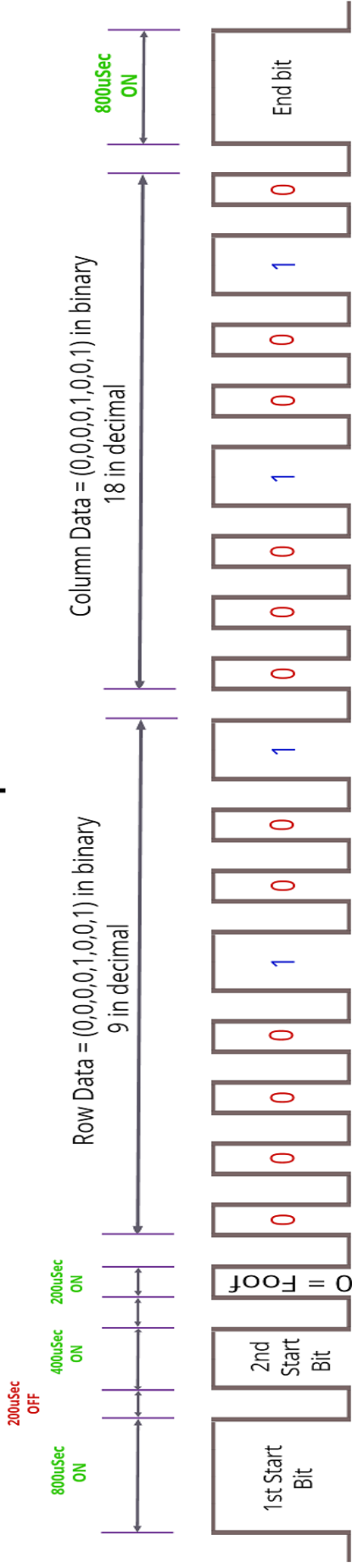
Actual Data that is transmitted through the IR LED



Data intercepted by the TSOP 1738

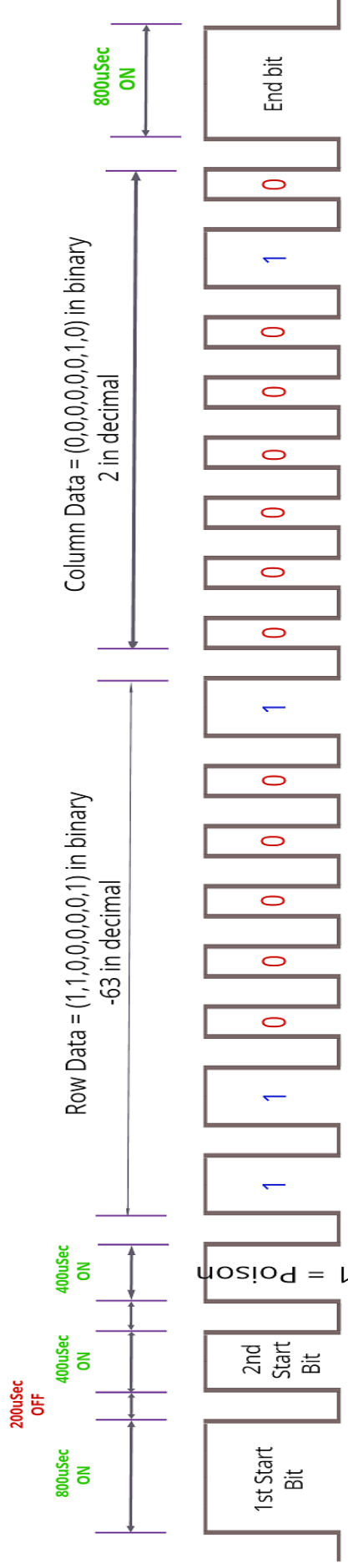


Example Data



Data interpretation : - The current node is a food node.

Relative location to the next node is 9,18 i.e Row 9 and column 18.



Data interpretation : - The node next to it is poison.

Location of the poison node is -63,2 i.e Row -63 and column 2.

The Arduino code for decoding data:-

Connect your TSOP to DIGITAL PIN 2 or 3

For any help regarding code please contact us

```
#define RECV_PIN 2 // Must be 2 or 3 for interrupt to work
#define INTERRUPT_PIN (RECV_PIN-2)
#define DATA_LENGTH 200
int i,count = 0;
long Start = 0, Stop = 0;
long data[60];
byte d[20];
void setup()
{
  pinMode(RECV_PIN, INPUT);
  Serial.begin(250000);
  data[0] = 25;
  delay(500);
  Serial.println("SETUP COMPLETE");
  attachInterrupt(INTERRUPT_PIN, startDataRecieve, FALLING);
}
void loop() {
  digitalWrite(13, LOW);
  delay(300);
  if (count){
    detachInterrupt(INTERRUPT_PIN);
    Serial.print("Total Input taken = "); // Comment these lines as they are only for decoding
    purpose
    Serial.println(count);
    for(i=0; i<count; i++) // Print what has been recieved
    {
      if(i%10 == 0)
        Serial.println();
      Serial.print(data[i]); // Prescale data, a way to remove errors by reducing resolution
      Serial.print(" ");
    }
    for(i=0; i<20; i++)
      d[i] = ((data[i]+70)/DATA_LENGTH); // Store the significant data
    Start = 0;
    decodeData(); // Now start decoding
    Serial.println("\n\t---READY---"); // Ready again to be flashed
    count = 0;
    attachInterrupt(INTERRUPT_PIN, startDataRecieve, FALLING);
  }
}

void decodeData()
{
  char row,col;
```

```

    if(d[0] == 4 && d[1] == 2)    //Checking for start bits, see the document for what data is
    recieved
        Serial.println("StartBit OK");
    if(d[2] == 1)                // Poision bit, 1 = No Poision and 2 = Poision
        Serial.print("No ");
        Serial.println("POISION");
    for(byte i=0; i<8; i++)      // Row decode, 1 is 0 and 2 is one
    {
        row <<=1;
        if(d[3+i] == 2)         // Bit wise shifting and decoding
            row |= 1;
    }
    for(byte i=0; i<8; i++)
    {
        col <<=1;
        if(d[11+i] == 2)        // Same for row data
            col |= 1;
    }
    Serial.print("Co-ordinate = (");
    Serial.print(row,10);
    Serial.print(" , ");
    Serial.print(col,10);
    Serial.println(") \n");
    if(data[19] == 4)            // You can check Start and End bits to validate your data
        Serial.println("Stop Bit Ok");
}

void fallTime()
{
    Start = micros();            // Counter started
    attachInterrupt(INTERRUPT_PIN, riseTime, RISING);    // Waiting for a RISING pulse on the
    interrupt pin
}

void riseTime()
{
    data[count] = micros() - Start;    // Record the HIGH time here
    attachInterrupt(INTERRUPT_PIN, fallTime, FALLING);    // Wait for fall interrupt
    count++;
}

void startDataRecieve()        // Called when data appears first time, initializes interrupt for
first rise time record
{
    attachInterrupt(INTERRUPT_PIN, riseTime, RISING);
    Start = micros();
}

```

END OF CODE

The Serial Monitor Output will look like this:-

```
COM6
---READY---
Total Input taken = 20
772  436  180  404  364  404  364  332  392  332
204  144  224  204  400  208  328  404  364  776  StartBit OK
No POISION
Co-ordinate = (-2 , 19)

---READY---
Total Input taken = 20
768  428  180  420  380  372  396  396  328  404
180  204  144  208  404  176  408  368  404  748  StartBit OK
No POISION
Co-ordinate = (-6 , 23)

---READY---
Total Input taken = 20
776  456  180  404  368  332  396  352  404  368
204  168  204  228  416  208  396  404  364  824  StartBit OK
No POISION
Co-ordinate = (-2 , 23)

---READY---
Total Input taken = 20
772  356  212  392  332  404  364  432  368  408
180  204  144  208  404  176  408  368  408  772  StartBit OK
No POISION
Co-ordinate = (-2 , 23)

☒ Autoscroll
Both NL & CR
250000 baud
```

For further help contact us personally and we will assist you:-

Name	Email	Phone Number (WhatsApp)
Ayush Agarwal	ayush240695@gmail.com	9051501840
Sourav Dokania	sourav.dokania25@gmail.com	9614989465

Thank you and best of luck at Srijan 2017.