

Simplex High-Speed Morse Coding with Ultra Low Power MSP430

Vibha Ravi, Vidhathri Udaya, Keshav Vaidyanathan Bharadwaj, Vivek Bettadapura Adishesha

Department of Electronics and Communication Engineering, RNS Institute of Technology, Visvesvaraya Technological University, Bengaluru, Karnataka, India

Email address:

vibha274@gmail.com (V. Ravi), vidhathrikrishna@gmail.com (V. Udaya), keshavbharadwaj@gmail.com (K. V. Bharadwaj), vivek.adishesha@gmail.com (V. B. Adishesha)

To cite this article:

Vibha Ravi, Vidhathri Udaya, Keshav Vaidyanathan Bharadwaj, Vivek Bettadapura Adishesha. Simplex High-Speed Morse Coding with Ultra Low Power MSP430. *American Journal of Embedded Systems and Applications*. Vol. 7, No. 1, 2019, pp. 26-40.
doi: 10.11648/j.ajes.20190701.14

Received: July 31, 2019; **Accepted:** September 2, 2019; **Published:** September 17, 2019

Abstract: This paper presents the design and implementation of a simplex communication model using high-speed Morse coding. Morse code is a dot and dash code used from the 19th century and prevails for highly secure applications. The objective of this paper is to design effective encoder and decoder circuits for the Morse code which draw less power. The model is implemented by using two microcontrollers – MSP430 for encoding and ATMEGA8 for decoding. The communication between microcontrollers and the end users are established using Bluetooth. The Morse code is transmitted through an optical medium as flashes of light. The light is flashed using an LED and detected using a phototransistor that constitute an optocoupler. The actuator and sensor pair in the optocoupler are varied by using combination of other actuators like IR diode, Laser and sensors like Photodiode. Each of these combinations have been analysed under the presence and absence of external light with varying distances and the resulting characteristics are plotted. This model is very effective for communication in noisy environments and eliminates the necessity of skilled personnel for decoding the message. It enhances the security and reduces the time required for the overall process. The model has been tested with information of various lengths and it has proven successful in performing the desired operation and delivering the correct output in all the cases.

Keywords: Simplex Communication, Optical Medium, Morse Coding, MSP430 Micro-controller, ATMEGA8 Micro-Controller, Bluetooth Communication, Photo-transistor

1. Introduction

Communication is the process of imparting or exchanging information through a medium. Simplex, Half-Duplex and Full-Duplex are the various methods of communication. Simplex communication is basically a one-way communication with only one sender which transmits the information. Examples of simplex communication include radio broadcasting, television broadcasting, communication between a computer and a printer etc.

The information is passed from the sender to the receiver via a path called the transmission medium. They can be either wired or wireless. An Optical medium is a type of wireless transmission medium through which electromagnetic waves can propagate. The light forms the electromagnetic carrier wave which is modulated to carry information. Here, the

sender uses an LED to transmit the modulated signal in the form of flashes of light of different duration. On the receiving side a photo-transistor is used to detect the flashes and decode the data based on their pattern.

Data travelling through various media is susceptible to access by unwanted third-party receivers. This leads to misuse of information and may also cause data manipulation. Encryption is the best way for secure transmission of confidential or personal data.

Morse Code is one such character encrypting scheme that encodes standardized text characters as two symbols of different duration called dots and dashes. It was developed in the early 19th century and was originally transmitted by telegraph. Each character is mapped to a unique code which is transmitted as a series of toggling tones, changing brightness levels or ticks [2]. It requires a skilled observer to decode the message. Learning

the code is difficult and time consuming. Military personnel still use Morse code to transmit high security information. Hence, we have devised a circuit that decodes the Morse code by analyzing the varying brightness levels.

International Morse Code

1. A dash is equal to three dots.
2. The space between parts of the same letter is equal to one dot.
3. The space between two letters is equal to three dots.
4. The space between two words is equal to seven dots.

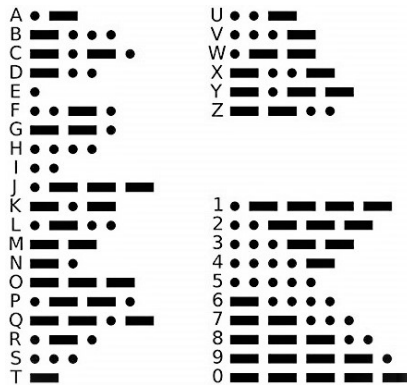


Figure 1. International Morse Code chart. The duration of a dash is three times that of a dot.

A micro-controller is a compact integrated circuit that governs a specific operation. There are several micro-controllers available in the market which are provided by

different vendors. Here we use two of them namely MSP430 from Texas Instruments and ATMEGA8 which is an AVR family micro-controller.

Key features of MSP430 are:

- a. Low to Ultra Low Power consumption
 - b. Several programmable power saving modes
 - c. On-Chip emulation and debugging features
- Key features of ATMEGA8 are:
- e. High Performance and low power consumption
 - f. Easy to handle and design
 - g. Single clock cycle execution and fully static operation.

As mentioned earlier communication can be wired or wireless. Bluetooth is a high speed, low power wireless technology designed to connect phones or other portable equipment together. Blue-Core 4 employs Radio frequency for communication. It makes use of frequency modulation to generate radio waves in the ISM band. Bluetooth networks use a master/slave model to control when and where the devices can send data.

Here, we use two sets of master/slave models which are:

- a. At the sending side, the sender's device (his mobile phone) acts as the master and the MSP430 Micro-controller acts as the slave.
- b. At the receiving end, ATMEGA8 acts as the master and the receiver's device (his mobile phone) acts as the slave.

The Blue-Core 4 (HC-05) module is interfaced with both the Micro-controllers. This module can be configured as either a master or a slave.

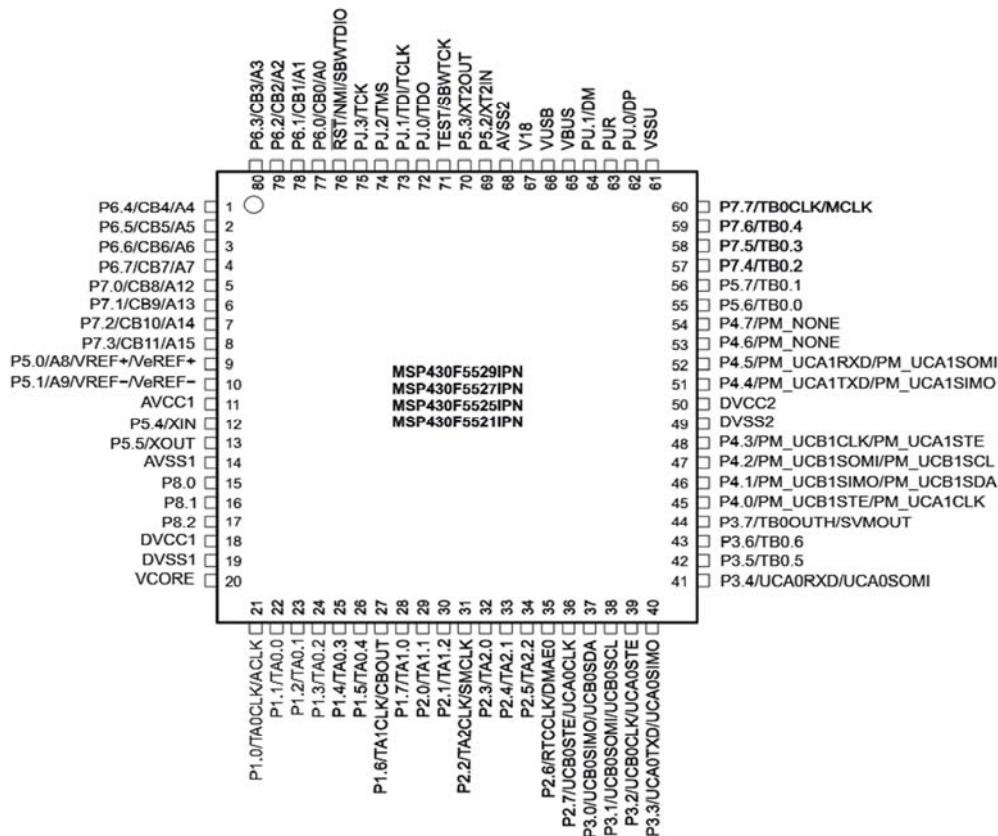


Figure 2. 80-Pin PN package of MSP430F5529 (Top View).

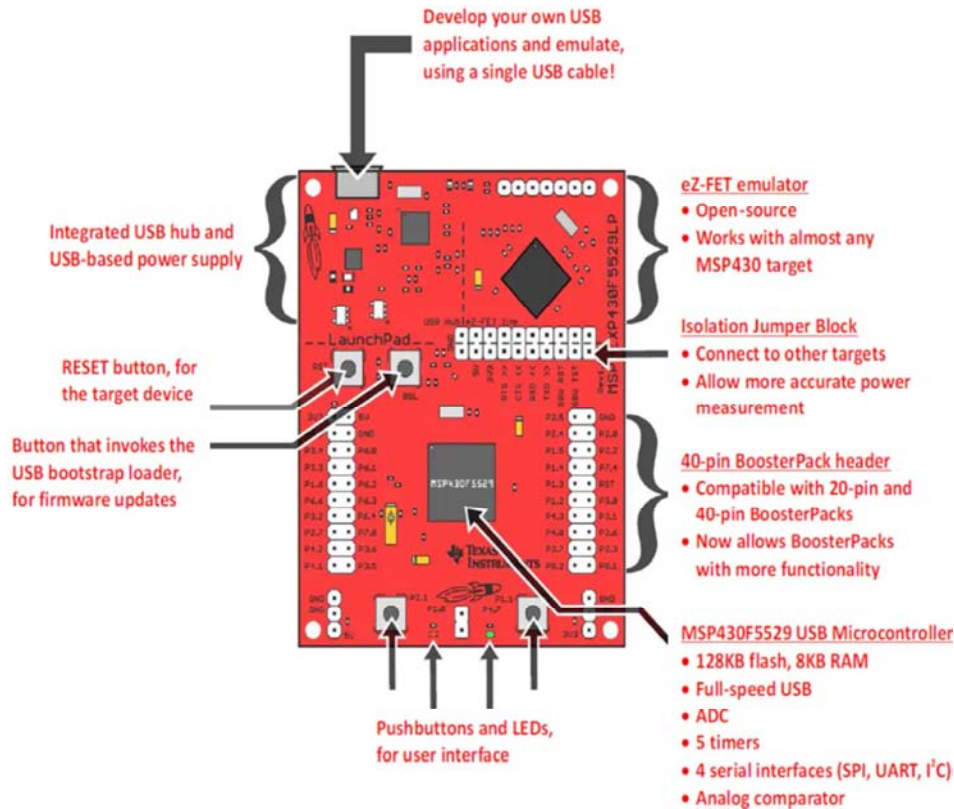


Figure 3. Components MSP430F5529L LaunchPad.

2. Hardware and Software Description

2.1. MSP430 Micro-Controller

The LaunchPad kit is from Texas Instruments

It is an inexpensive, simple micro-controller kit with an on-board emulation for programming and debugging, as well as buttons and LED's for simple user interface [1]. Few features of it are:

- USB 2.0-enabled 16-bit MCU.
- Up to 25 MHz CPU speed.
- 12KB Flash and 8KB RAM.
- 12-bit SAR ADC.
- 40 pin BoosterPack ecosystem.
- 5V and 3.3V through a efficient DC/DC converter.
- Serial communication interfaces like UART, I2C, SPI are available.

Figures 2 and 3 show pin-out diagram and the component diagram of the MSP430F5529 microcontroller respectively.

2.2. ATMEGA8 Micro-Controller

In 1996, Advanced Virtual RISC (AVR) Micro-controller was produced by the "Atmel Corporation"[8]. Its features are:

- It has 8Kb of Flash program memory.
- SPI, USART, TWI.
- It has an Analog Comparator and a 10-bit ADC.
- 32 * 8 General Purpose Working Registers.
- On-Chip 2-cycle Multiplier.

- Five sleep modes.
- Operating voltage: 4.5 – 5.5V.

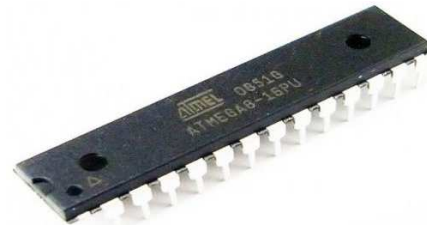


Figure 4. ATMEGA8 Micro-controller.

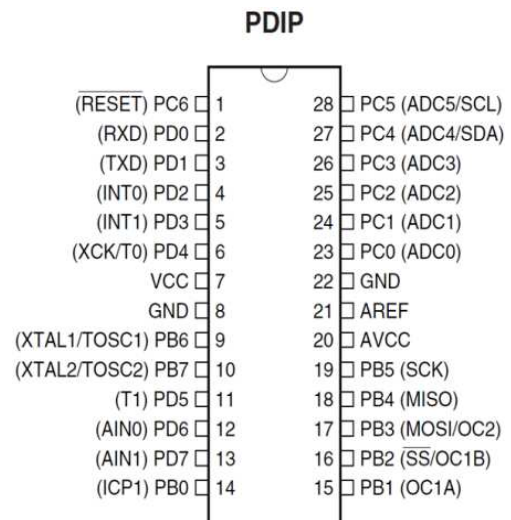


Figure 5. Pin Configuration of ATMEGA8 Micro-controller.

2.3. Blue-Core4 in HC-05 Bluetooth

HC-05 is a class-2 Bluetooth module which houses Blue-Core 4 BC417 chip for wireless communication. Has master or slave configuration.

It has a range up to <100 meters which depends on the environment. IEEE 802.15.1 standardized protocol is used through which one can build wireless Personal Area Network (PAN). It uses Frequency Hop Spread Spectrum (FHSS) radio technology to send data over air. [6]

It communicates with Micro-controller using serial port (USART). Few specifications of the module are:

- Operating frequency: 2.4GHz ISM Band.
- Modulation: GFSK Emission power: <4dBm.
- Security: Authentication and encryption.
- Power Supply: +3.3V DC 50mA.
- Speed:
Asynchronous: 2.1 Mbps (max) / 160 Kbps.
Synchronous: 1 Mbps.



Figure 6. Blue-Core 4 chip on HC-05 Module.

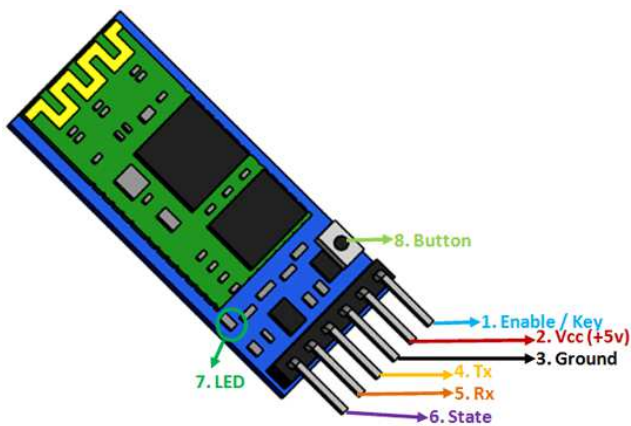


Figure 7. Pin-out of the HC-05 Bluetooth Module.

2.4. Photo-transistor (SK1KL3B)

A photo-transistor is a three-layer semiconductor device with a light sensitive Base region. It detects the flashes of light and alters the current flowing between its Emitter and Collector relative to the intensity of light it receives.

The ST1KL3B photo-transistor functions as a sensor at the receiving end. It is a high sensitivity NPN silicon transistor mounted in durable, hermetically sealed TO-18 metal can which can provide years of reliable performance even under demanding conditions such as outdoor use.



Figure 8. Image of the ST1KL3B Photo-transistor.

2.5. Code Composer Studio (CCStudio)

The MSP430F5529 Micro-controller has been programmed using CCStudio version 8.1.0. It is primarily designed as for embedded project design and low-level JTAG based debugging. The highlights of this software are:

- Single integrated development environment for all TI processors.
- Integrates source code editor, debugger, optimizing C/C++ compiler, code profiler etc.
- Seamless Host-Target communication using Real Time Data Exchange..

2.6. Arduino IDE

Arduino integrated development environment is a cross-platform application written in the programming language Java. Here we write the C program for the ATMEGA8 Micro-controller. The program is dumped onto the chip by making ATMEGA8 as ISP. The steps involved in programming the ATMEGA8 using Arduino IDE are as follows:

- Add ATMEGA8 support to Arduino IDE using Board Manager.
- Program the Arduino as ISP: Upload the sketch from the path - File → Examples → 11. Arduino ISP → Arduino ISP.
- Burn the Boot-loader as follows:
 - Make the following connections:
 - Go to Tools→Board and select ATMEGA8.
 - Go to Tools→Programmer and select 'Arduino as ISP'.
 - In Tools menu select 'Burn Boot-loader'.
- Upload the sketch: Press and hold the Reset button and click on upload. Release the Reset button when it shows 'Uploading'.

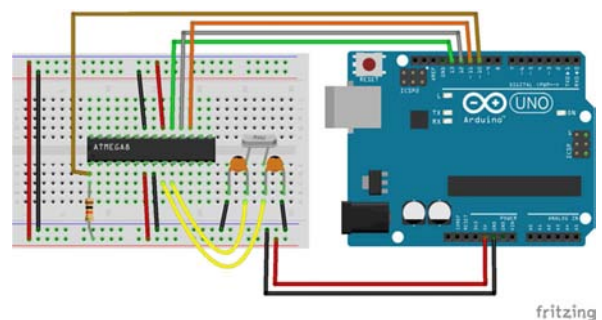


Figure 9. Circuit connections for uploading program to ATMEGA8 Micro-controller.

3. Implementation

3.1. Software Implementation

The program for the entire project is split into two parts. The sender i.e. MSP430 Microcontroller is programmed using CCStudio version 8.1.0 and the receiver i.e. ATmega8 is programmed using Arduino IDE.

The overall functioning of the model is depicted in the flow chart below.

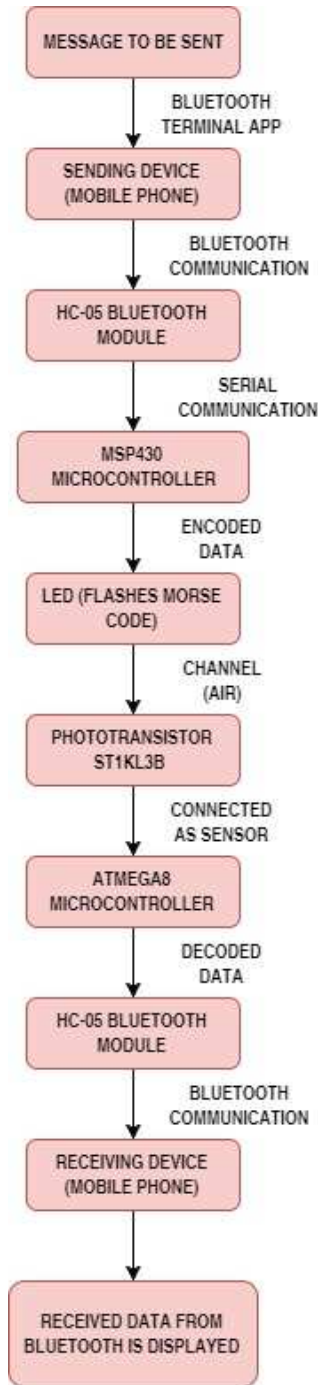


Figure 10. Overall functioning.

1. The sending side

The program for MSP430 Microcontroller has been written in Embedded C Language. It consists of the following parts.

- Disabling Watchdog Timer
- Interfacing the Bluetooth module
- Receiving the input data
- Converting the data into corresponding Morse code
- Flashing the Morse code

The main structure of the program is depicted in the flow chart given below.

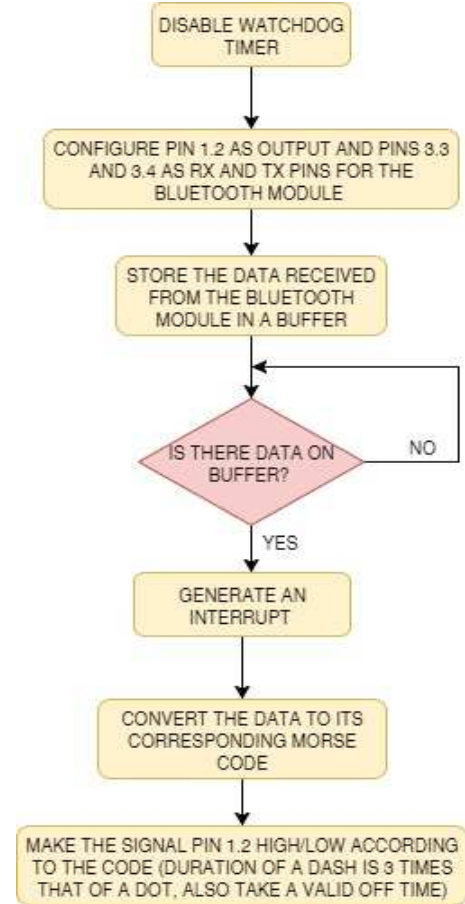


Figure 11. Working of Transmitter.

Disabling the Watchdog timer in MSP430 is simple and straight forward. The below command disables the Watchdog timer.

```
WDTCTL = WDTPW | WDTHOLD;
```

The data is received from the sender's device via Bluetooth. The Bluetooth module is interfaced with the MSP430 by connecting it to the TX and RX pins (P3.3 and P3.4). The data received is read from the RX pin and stored in the buffer.

```

P3SEL |= BIT3+BIT4;           // P3.4,3.3 = USCI_A1 TXD/RXD
UCA0CTL1 |= UCSWRST;          // Put state machine in reset
UCA0CTL1 |= UCSSEL_2;          // SMCLK
UCA0BR0 = 109;                 // 1MHz 115200 (see User's Guide)
UCA0BR1 = 0;                   // 1MHz 115200
UCA0MCTL |= UCBRS_1 + UCBRF_0; // Modulation UCBRSx=1, UCBRFx=0
UCA0CTL1 &= ~UCSWRST;          // Initialize USCI state machine
UCA0IE |= UCRXIE;              // Enable USCI_A1 RX interrupt
__bis_SR_register(LPM0_bits + GIE); // Enter LPM0, interrupts enabled
__no_operation();               // For debugger
  
```

Each character stored in the buffer is converted to its corresponding Morse code. Let us consider letter 'a' as the input. The corresponding code for 'a' is a dot and a dash. It is converted in the following way:

```
case 'a':    dot(1);           //1 dot symbol
            dash(1);          //1 dash symbol
            delay(13000);
            break;
```

The delay is generated using the Timer B present in the MSP430F5529. The capture compare mode helps in the generation of delay and is implemented as follows.

```
void delay(int x)
{
    TB0CCR0 = x;           //capture compare reg value
    TB0CTL = TBSSSEL_1 + MC_2; // mode-2 , ACLK
    TB0R = 0;              // TAR register update
    TB0CCTL0 = CCIE;
    while(TB0R < 2000)
    {
        __bis_SR_register(LPM0_bits | GIE); // enter LPM
        __no_operation();
    }
    TB0CTL = 0;           // reset timer A0 control register
}
```

The LED on which the output Morse code is flashed is connected to P1.2. If the code contains a dot, the output pin is held high for 1 time unit and if the code contains a dash the pin is held high for 3 time units.

```
void dot(int n)           // n is the number of dots
{
    volatile int y=0;
    while(y<n)
    {
        delay(10000);     // on for 4000, off for 10000
        P1OUT=0x04;
        delay(4000);
        P1OUT=0x00;
        y++;
    }
}

void dash(int m)          //m is the number of dashes
{
    volatile int z=0;
    while(z<m)
    {
        delay(10000);     // on for 12000 and off for 10000
        P1OUT=0x04;
        delay(12000);
        P1OUT=0x00;
        z++;
    }
}
```

2. The Receiving Side

The program for ATMEGA8 Microcontroller is written on the Arduino IDE and is programmed through ISP. The major parts of the code are

- Enable serial communication.
- Determine whether the incoming symbol is a dot or dash.
- Translate the symbols to the corresponding text.
- Send the data to the receiver's phone.

The flow of execution of the program is depicted in the following chart:

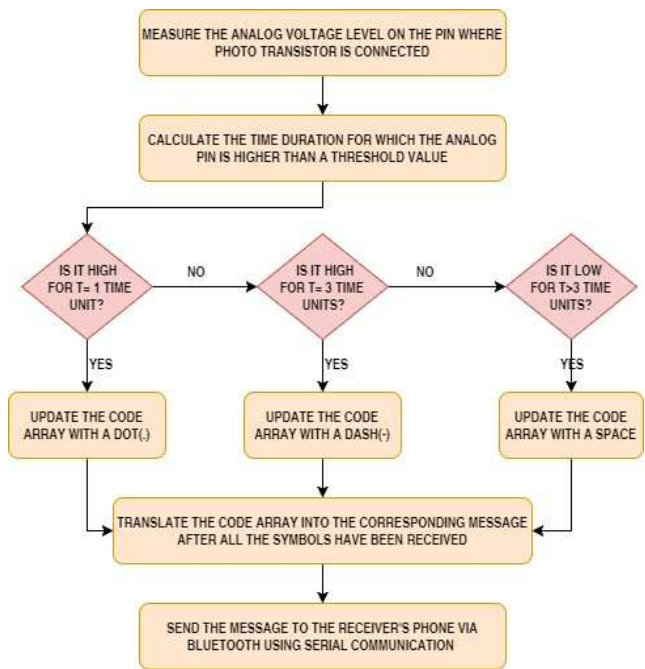


Figure 12. Working of Receiver.

The serial communication is enabled with a baud rate of 9600. The Bluetooth module uses serial communication.

Serial.begin (9600);

The symbol is determined by measuring the duration for which the analog input pin to which the sensor (photo-transistor) is connected remains high. The duration is determined by taking the time difference between the on and off state of the pin. The time has been measured using the `millis()` function. A code array is created and updated with a dot or a dash depending on whether the signal pin is high for 1- or 3-time units respectively.

```

t1=millis();
while(analogRead(A0)>=150);
t2=millis();
time=t2-t1;
if(time>5)
{
    if(time<25)
    {
        code[len]='.'; //Update a dot
        len++;
    }
    else if(time>25)
    {
        code[len]='-'; //Update a dash
        len++;
    }
}
  
```

Similarly, if the signal pin is LOW for a considerable amount of time, update the code array with a space.

After all the symbols have been received, the code array is translated into the corresponding text message. An array containing the mapped Morse data is constructed and used for this purpose. Each symbol is compared with the value in the array and the character is determined. If none of the values match, unknown is printed.

The mapping array is given as:

```
String decode[29] = {"-", "-", "-", "-", "-", "-", "-", "-", "-", "-", "-", "-", "-", "-", "-", "-", "-", "-", "-", "-", "-", "-", "-", "-", "-", "-", "-", "-"}, //Mapping array
```

Each symbol is compared one at a time. So, each symbol is copied into a temporary 'copy' string as follows:

```
while(code[i]!=' ') //Copy till a space is found
{
    copy+=code[i];
    i++;
}
```

The symbol is translated as follows:

```
while (decode[j] != "2") // comparing input code with Letters array
{
    if (decode[j]==copy)
    {
        Serial.print(char('A' + j));
        break;
    }
    j++;
}
j++;
if(decode[j+1]==copy)
{
    Serial.print(" ");
}
else if(decode[j]==copy)
{
    Serial.println("\n Incoming");
}
else if(decode[j+2]=="2")
{
    Serial.println("Unkown"); // doesn't match any Letter, error
}
copy="";
j=0;
```

The Serial.print command writes the parameter on the serial buffer. The Bluetooth module connected to the serial port sends the data on the buffer to the receiving device. [3, 7]

3.2. Hardware Implementation

3.2.1. Circuit Diagrams

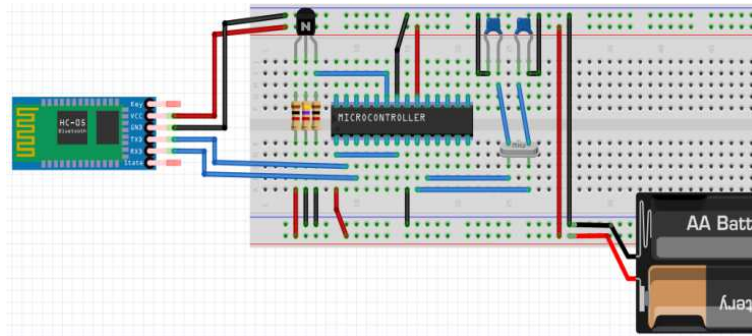


Figure 13. Circuit of receiver with Photo-transistor.

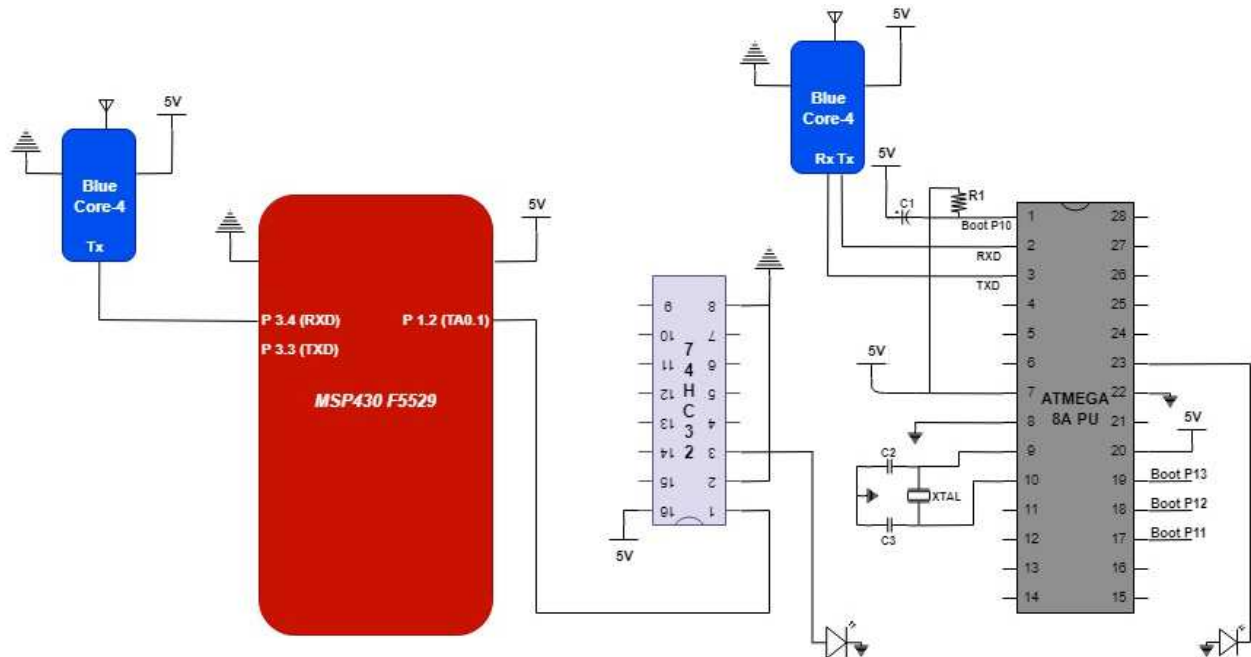


Figure 14. Schematic of the setup.

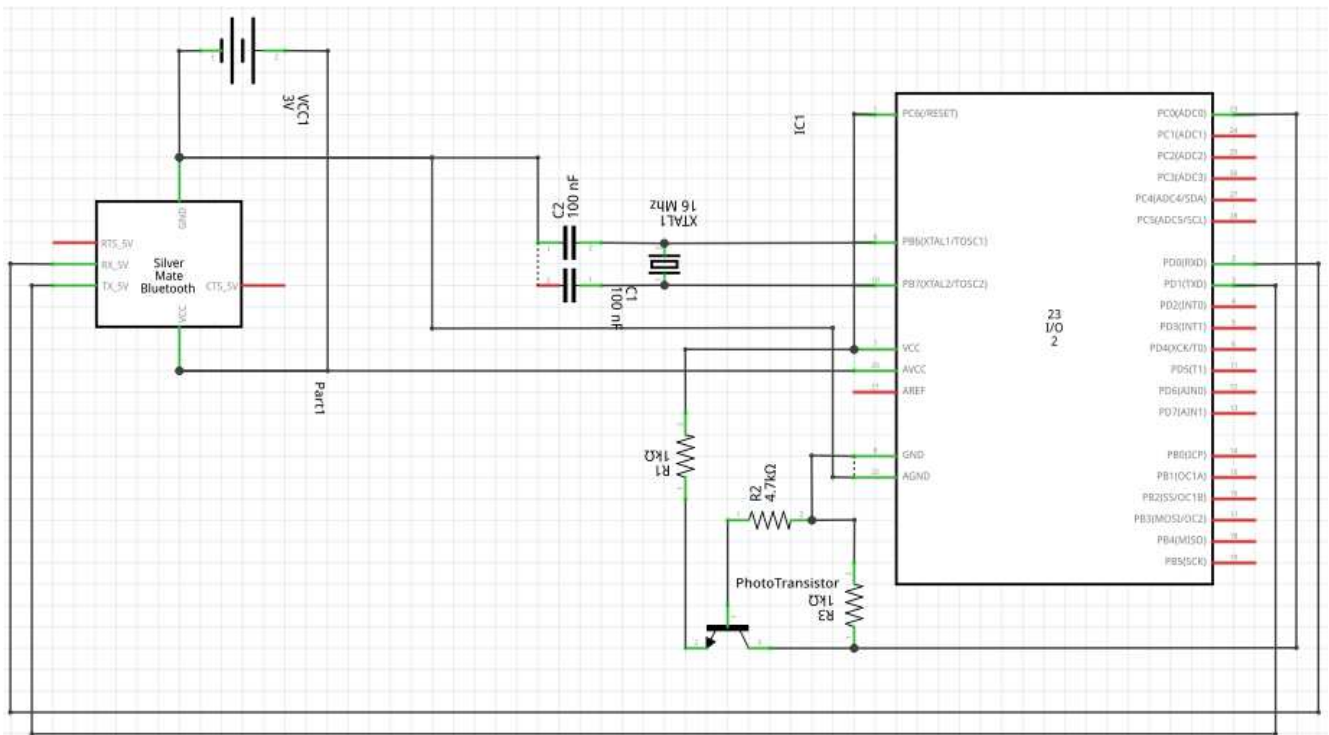


Figure 15. Schematic of ATMEGA8 receiver.

3.2.2. *Outputs*

methods can be used instead of Blue-Core4. HelloWorld messages are sent. The character ‘/’ is for indicating start and end of message flags. At the receiver it converts to ‘Incoming’ message text. As seen in Figure 16.

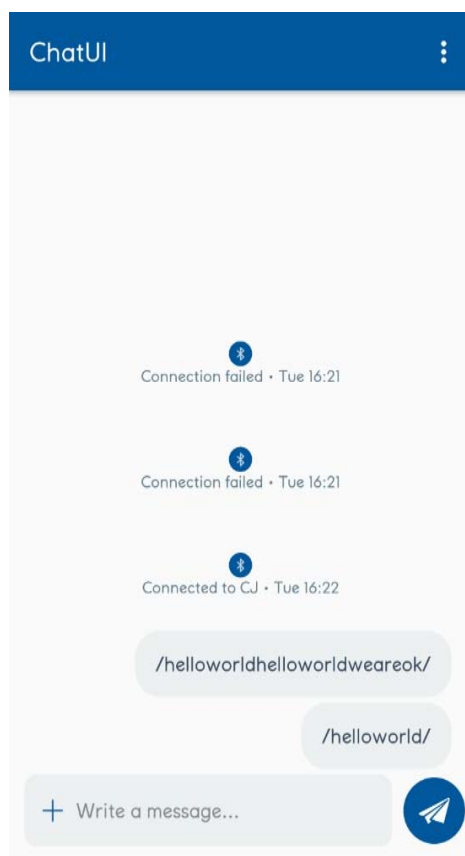


Figure 16. Transmitted message from sender device.

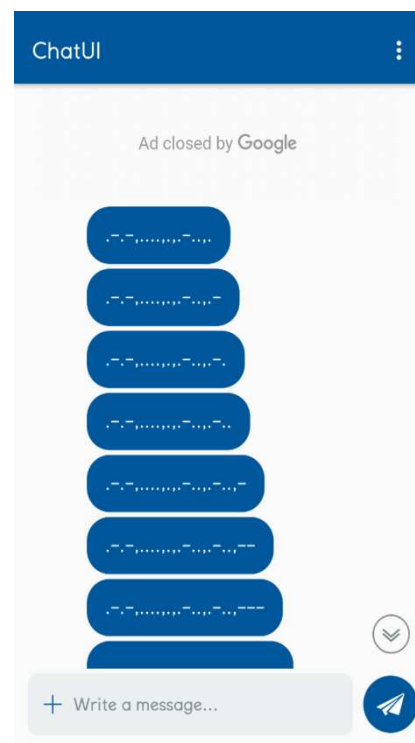


Figure 17. Morse code received.

Transmitter end, sending the messages to MSP430, other

The Morse codes received at the receiver end or ATMEGA-8 end. The decoding can be done with the help of

International Morse code given above. The codes are comma separated and the code of each letter is separated from each other. During the decoding process the commas are omitted. As in Figure 17.

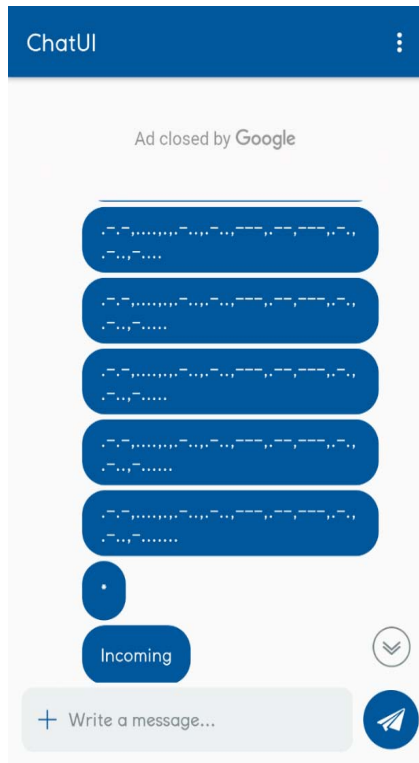


Figure 18. “HELLO WORLD” message in morse.

Figure 18. shows the entire received code for the message “HELLO WORLD”.

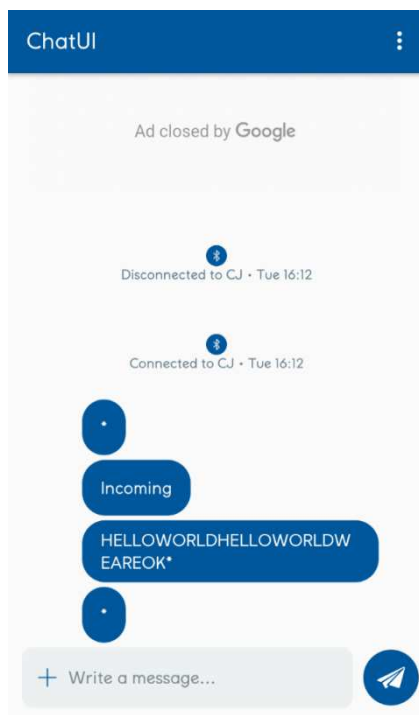


Figure 19. Translated message and received at the device.

The decoded message in ASCII text is seen in Figure 19. The Morse codes are then translated to English Alphabets and transmitted via Blue-Core4 to the reader/ receiver device.

3.2.3. Hardware Setup

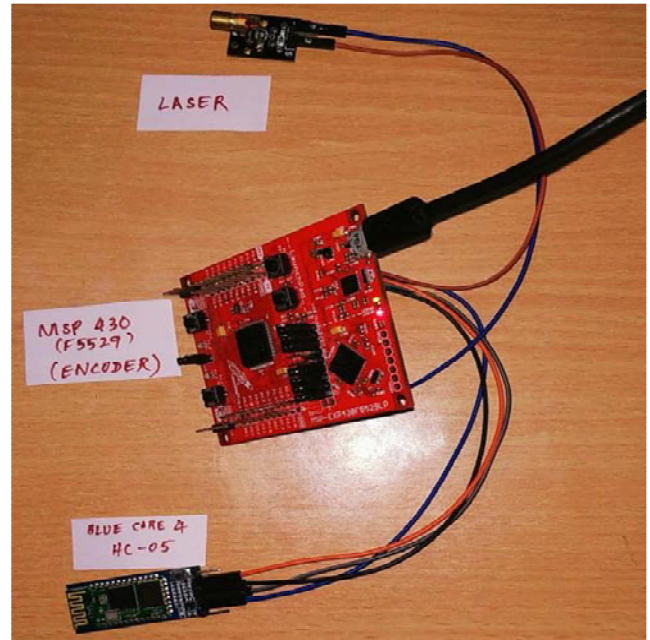


Figure 20. Transmitter setup.

Figure 20 shows the setup at the Transmitter end. It comprises of MSP430F5529, BlueCore4 module and a laser diode.

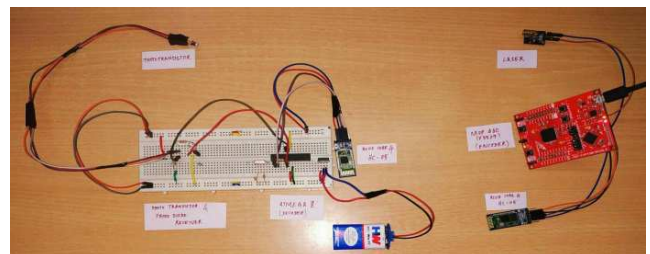


Figure 21. Overall Setup of the Experiment.

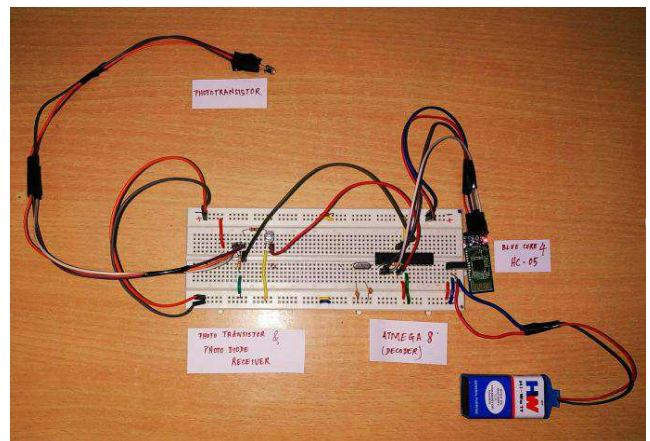


Figure 22. Receiver circuit with Atmega8 & Phototransistor.

Figure 21 displays one of the variations of the setup where the MSP430, ATEPGA8 and Blue-Core4 are used. The receiver set up, houses the Phototransistor as shown in Figure 22.

3.2.4. Test Results

In the setup LED is the transmitter and Photodiode is the

receiver. Two environments are chosen, with External light of fluorescent in nature and without External lights. The analysis shows that the external light slightly helps in the detection of the data without errors from further distances. Figure 23.

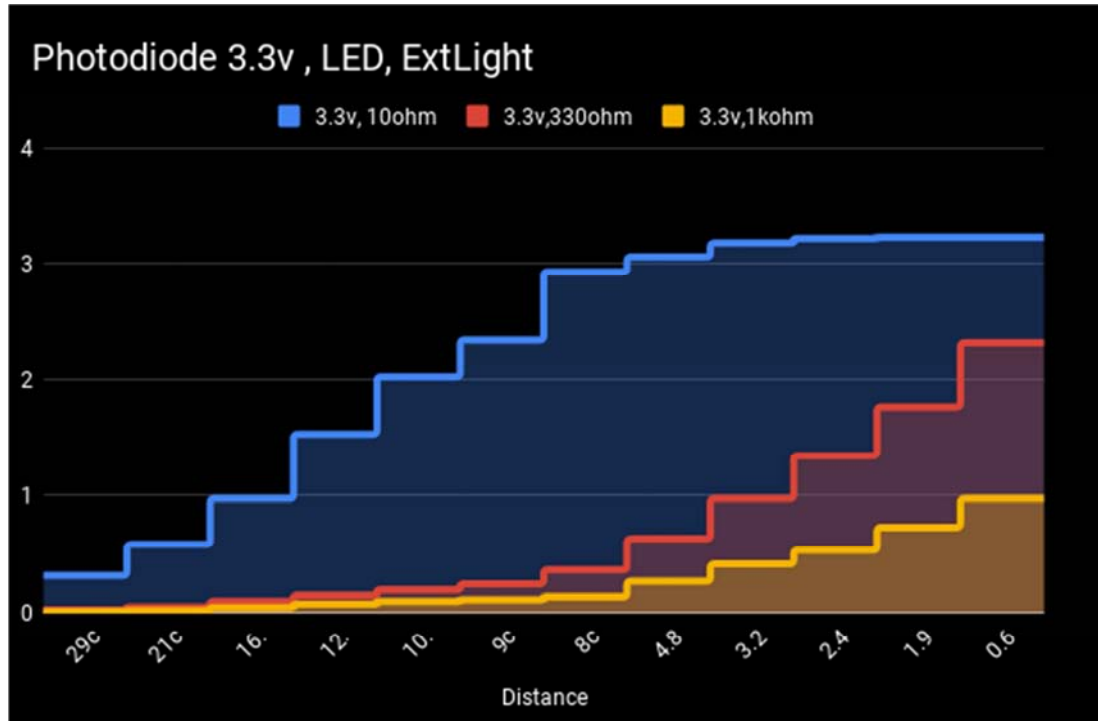


Figure 23. Voltage characteristics of Photo-diode receiver with LED source and External light.

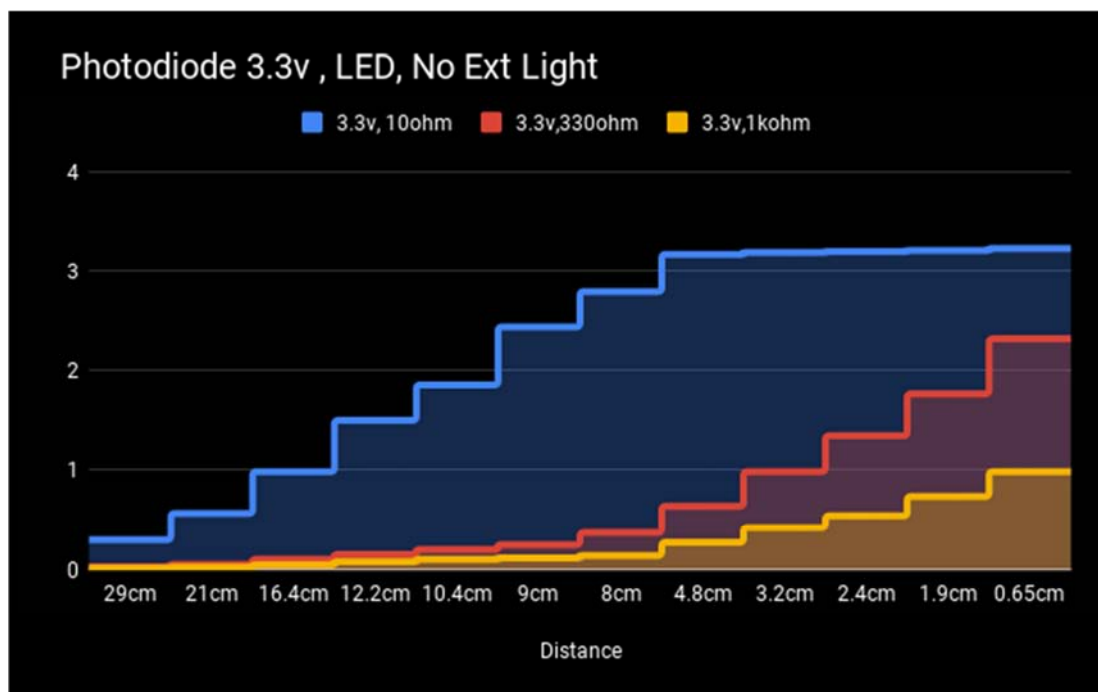


Figure 24. Voltage characteristics of Photo-diode receiver with LED source and No External light.

The analysis is also done on the intensity or voltage input to the LED. As the resistance of the transmitting led is reduced detection of data starts from longer distances. We

observe that at around 10 Ohms we have more voltage at Photodiode than 330 Ohms at the same distance. Figure 24.

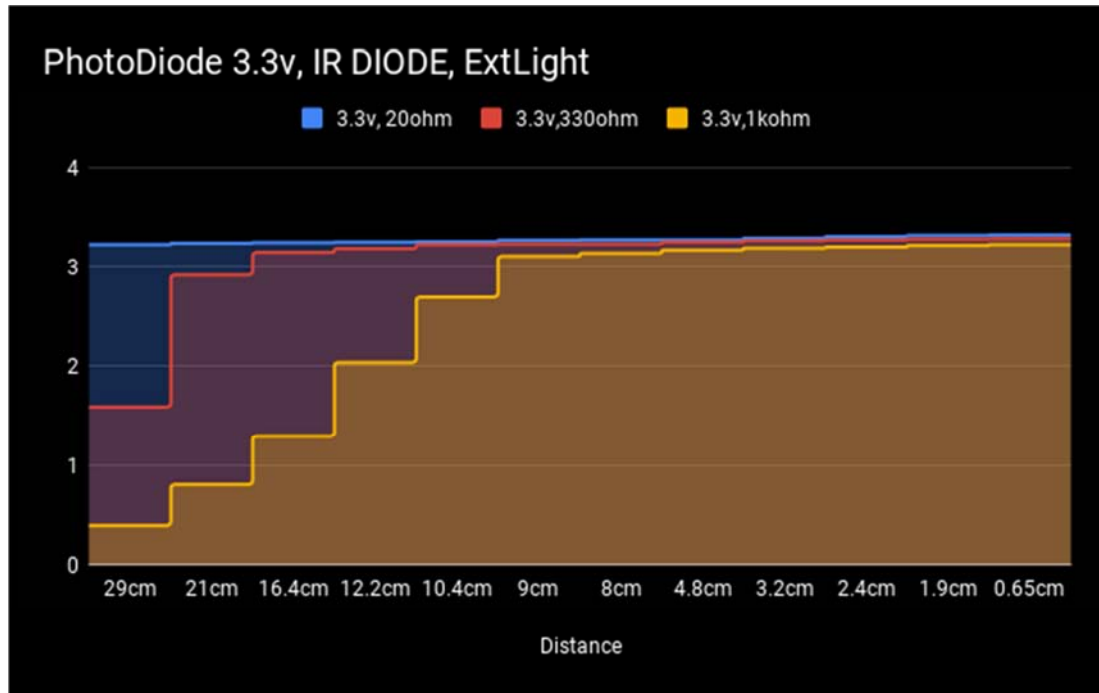


Figure 25. Voltage characteristics for Photo-diode receiver and IR source.

Setup for Figure 25 analysis includes an IR Diode and a Photo-diode for transmission and receiving respectively. IR Diode gives better saturation of voltage towards 3.3v as they are brought closer. This reduces error in data transmission.

As the resistance of the diode input is reduced the voltage saturates faster. IR has more field of operation, not requiring Line of Sight.

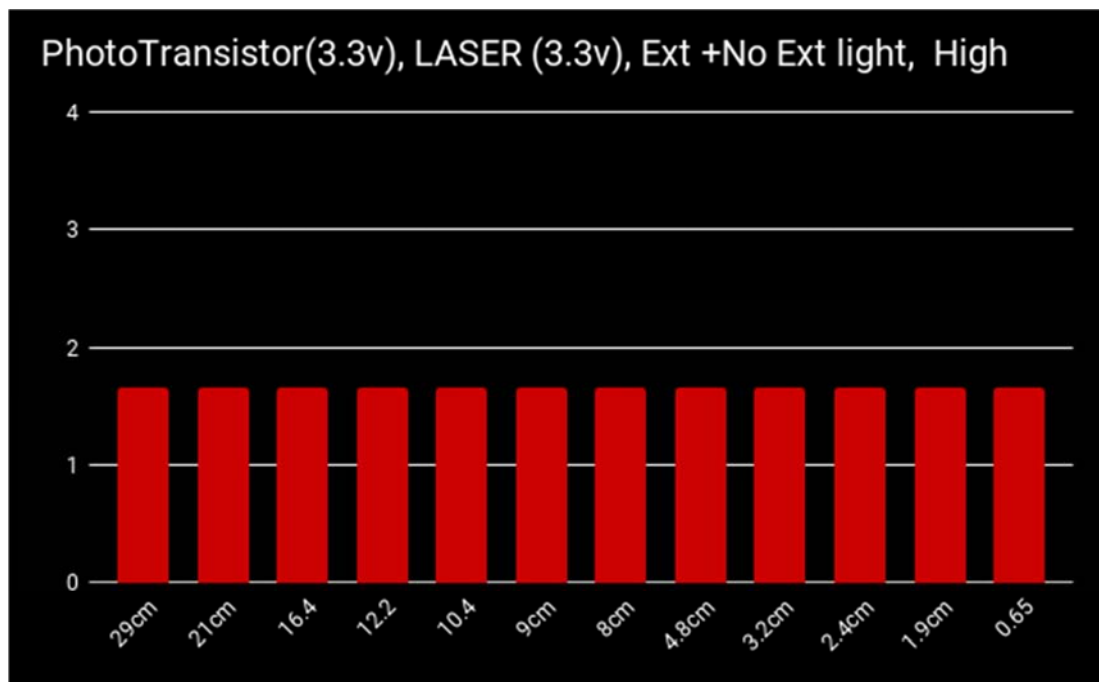


Figure 26. Photo-transistor receiver and Laser source, received voltage characteristics.

Figure 26 setup uses a 3.3v Red Laser. The analysis depicts that the laser's distance is not a factor for the voltage at the receiving Photo-transistor. The major disadvantage of using the laser in air medium is, in the region of high intensity is focused is very small. Alignment of the receiver

with the laser has to be precise. Phototransistor allows for the gain improvement, unlike the Photodiode counterpart. This advantage keeps the intensity of output as the same even as the laser moves further apart. As distance is not being a factor it can be used widely.

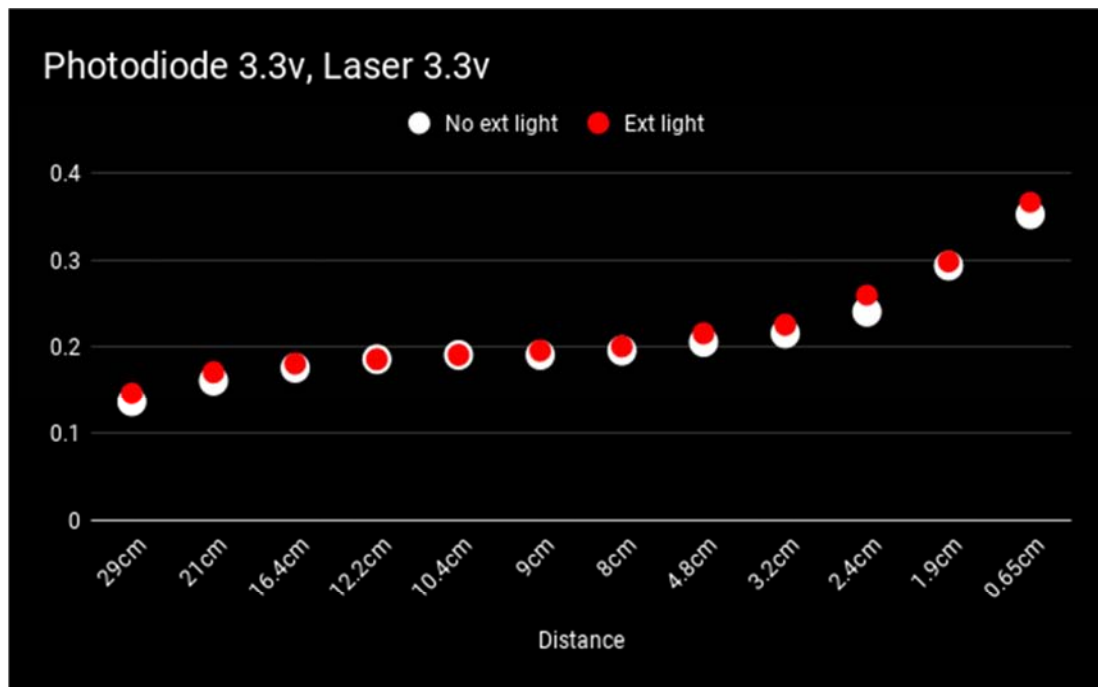


Figure 27. Curve for voltage at Photo-diode for Laser source.

In Figure 27 circuit setup is as follows, Laser and Photodiode as transmitter and receiver respectively. The external light influences the voltage at the Photodiode from the laser intensity, but very marginal. This is the boots by

using a Laser, no surrounding light effect unlike using a LED, also the intensity variation is minimal compared to the LED.

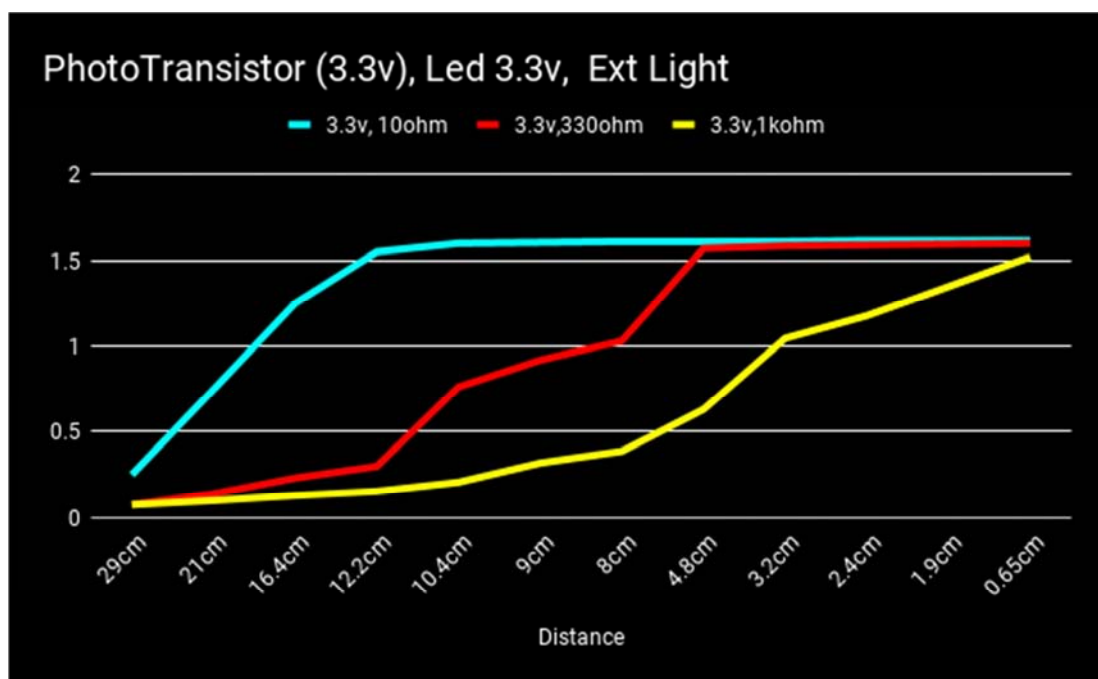


Figure 28. Curve for voltage at phototransistor with LED as source with external light.

The setup uses Phototransistor at receiver and LED at the transmitter. As the resistance at the input drops the voltage saturates quickly even when the transmitter is further away from the receiver. This setup is analyzed under external light sources. Figure 28 is the analysis of this setup.

With no external light sources affecting the setup, the voltage saturation for the intermediate resistance level has changed. Now it saturates at a further distance from the receiver unlike in situation with external light sources as in Figure 29.



Figure 29. Curve for voltage at phototransistor with LED as source without external light.

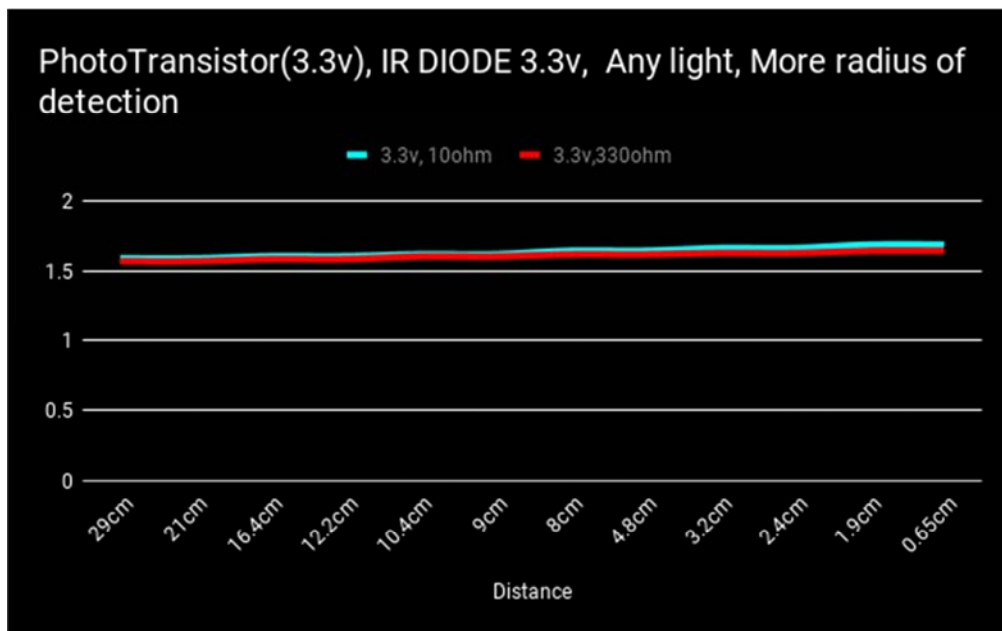


Figure 30. Voltage characteristics of phototransistor with IR Diode as source.

In Figure 30 setup, we use a Photo-transistor to receive instead of Photo-diode. This improves the results as the gains from the transistor has reduced the distance factor on the voltage of the received input. For different resistances, at the input diode the voltage saturates quickly, hence the resistance also plays an insignificant role.

The power consumption at different resistance levels at the input is analyzed. Here LED of white spectrum is consuming lesser power at low resistances compared to an IR Diode, this is depicted in Figure 31.

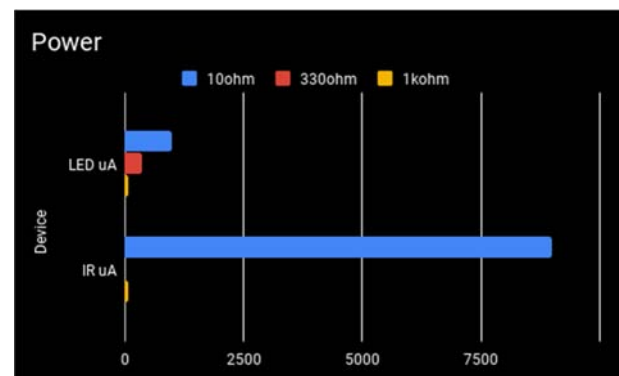


Figure 31. Power analysis for varying input resistance.

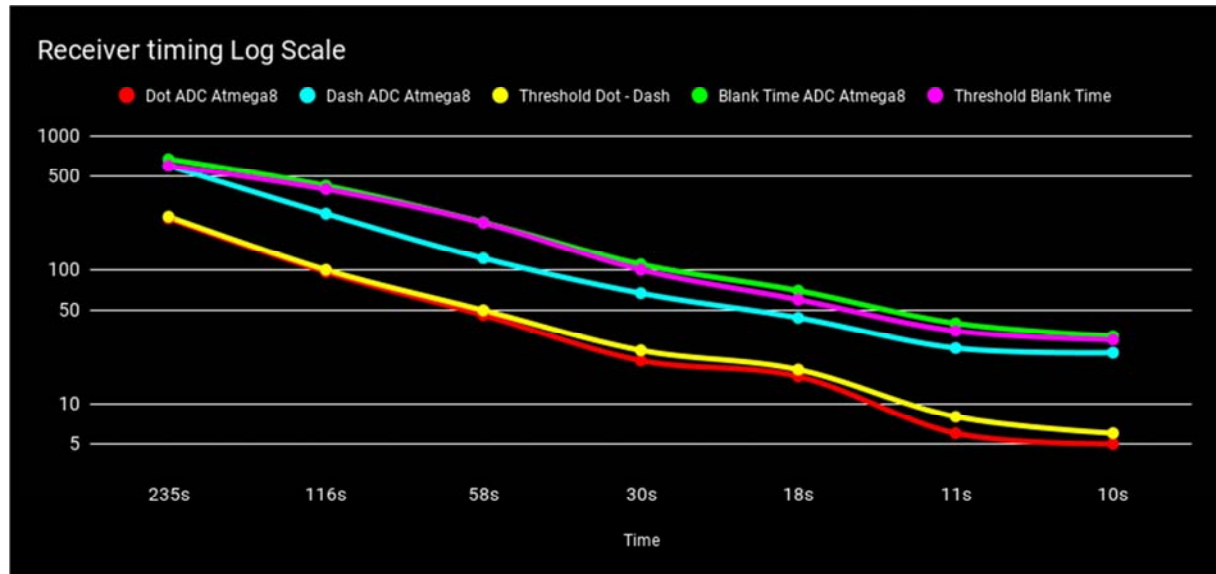


Figure 32. Receiver timing analysis curves.

The time axis represents the average time required for transmitting 100 characters of alphanumeric in Figure 32 & Figure 33.

The Receiver end is the Atmega 8 ADC. the values is the time of dot duration and dash duration values in milliseconds.

It has been plotted on a log scale to have better visualization. The threshold for Dot-Dash is the deciding region whether the symbol was a dot or dash. Blanking interval is the time during which no symbol is sent and this is considered as a space between words. Figure 32 shows the receiver end analysis.

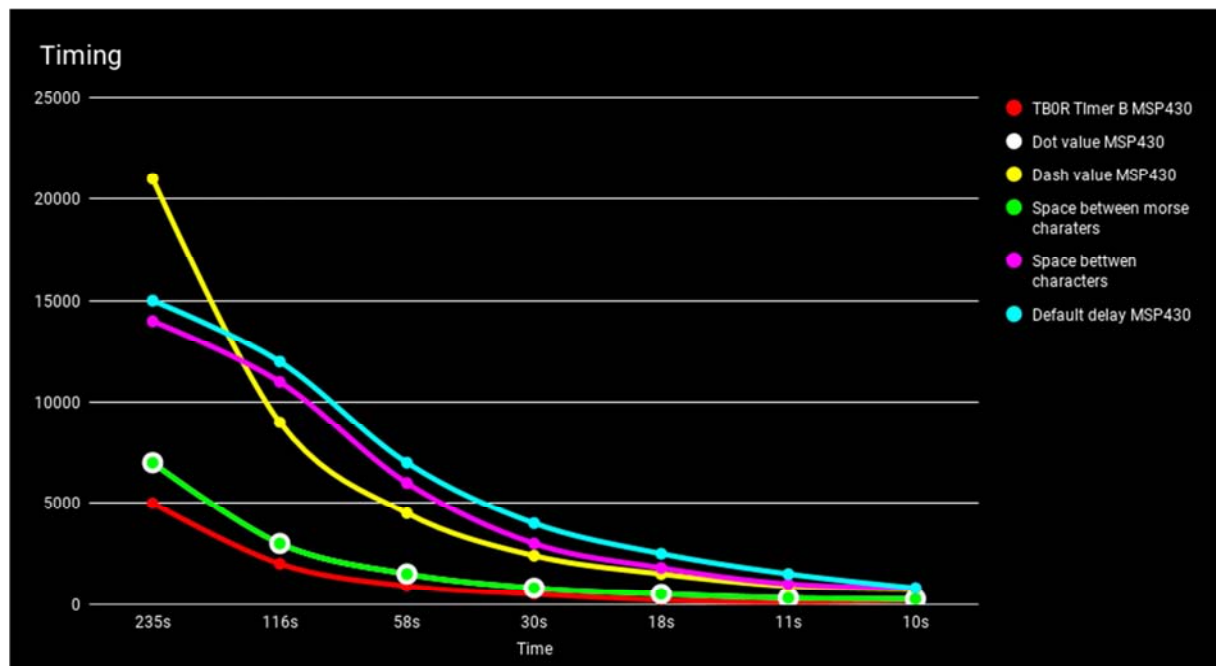


Figure 33. Curves for timing analysis at transmitting end.

The transmitting end is an MSP430 and can generate stable symbols in Ultra-Low Power. This is plotted in a linear scale and we can see how the values for faster transmission of 100 characters drop exponentially. TB0R is the Timer-B register of MSP, this helps in generation delays without consuming CPU cycles and power. Figure 33 shows the transmitter end analysis.

4. Applications

The Indian Navy uses various communication systems like ACCS (Advanced Composite Communication System) for exchange of information between two ships or ship and the shore. These systems have a Morse transmission switch. In emergency situations, when all other communication systems

fail or in special situations involving high security, transmission is switched to Morse manually.

Deep underground mines need communication interface for smooth work progress. Due to disturbance of radio signals underground, optical communication via Morse can be used. The miners can learn it very easily and be supported with short range RF devices connected to them.

5. Conclusion and Future Scope

This paper explains the successful implementation of a simplex communication model that operates on Morse code using an ultra-low power MSP430F5529 Microcontroller. Text messages could be easily sent from one device to another at high speeds. Using Morse code ensures high security as only a skilled person can decode it. So, the efforts of any intruder tapping the message will be unsuccessful. On the contrary, the decoder which is designed enables the intended receiver to obtain the decoded message without efforts as it is automatically translated. This method of communication has got dual benefit of good bandwidth efficiency and low transmission power as compared to the other complex coding schemes. Also, using the MSP430 microcontroller with its wide range of LPM (Low Power Modes) reduces the power even more. It is comparatively more immune to interference, both natural and man-made. It has a great scope in the fields of aviation to communicate with the base station, in navy to communicate with different ships, radio communication like the Amateur Radio etc. Recently it has proved to be an important communication tool for the people with various disabilities to communicate. [4]

References

- [1] Naveen Kumar Uttarkar, Raghavendra Rao Kanchi, "Design and Development of a Low-Cost Embedded System Laboratory Using TI MSP430 Launch Pad", American Journal of Embedded Systems and Applications. Vol. 1, No. 2, 2013, pp. 37-45. doi: 10.11648/j.ajes.20130102.12.
- [2] Manisha Barse, Rodney Manuel, "Morse Code - A Security Enhancer", International Journal of Science and Research (IJSR), ISSN (Online): 2319-7064, Volume 5 Issue 8, August 2016.
- [3] N. S. Bakde, A. P. Thakare, "Morse Code Decoder - Using a PIC Microcontroller", IJSETR, ISSN: 2278-7798, Volume 1, Issue 5, November 2012.
- [4] Ka, Hyun & Simpson, Rich. (2012). "Effectiveness of Morse Code as an Alternative Control Method for Powered Wheelchair Navigation".
- [5] Gaspar, Pedro & Espírito Santo, António & Ribeiro, Bruno. (2010). "MSP430 microcontrollers essentials - A new approach for the embedded systems courses: Part 2 - System and peripherals".
- [6] Pratibha Singh, Dipesh Sharma, Sonu Agrawal, "A Modern Study of Bluetooth Wireless Technology", International Journal of Computer Science, Engineering and Information Technology (IJCEIT), Vol. 1, No. 3, August 2011.
- [7] Anisha Cotta, Naik Trupti Devidas, Varda Kalidas Naik Ekoskar, "WIRELESS COMMUNICATION USING HC-05 BLUETOOTH MODULE INTERFACED WITH ARDUINO", ISSN: 2278 – 7798 International Journal of Science, Engineering and Technology Research (IJSETR) Volume 5, Issue 4, April 2016.
- [8] Kunikowski, Wojciech & Czerwiński, Ernest & Olejnik, Pawel & Awrejcewicz, Jan. (2015). An Overview of ATmega AVR Microcontrollers Used in Scientific Research and Industrial Applications. PAR. 19. 15-20. 10.14313/PAR_215/15.
- [9] M. Rezaei, M. S. Park, C. L. Tan and H. Mohseni, "Sensitivity Limit of Nanoscale Phototransistors", arXiv: 1704.05987v1 [physics.ins-det] 20 Apr 2017.
- [10] Lidyawati, Lita & Ramadhan Darlis, Arsyad & Jambola, Lucia. (2016). Implementation of Speech Simplex Communication using Visible Light, Conference: The 2016 International Symposium on Electronics and Smart Devices, At Institut Teknologi Bandung, West Java, Indonesia.
- [11] Khayal S. Chauhan, Kirit. R. Bhatt, "Portable Multi-Purpose Instrument Using MSP430 on Android Platform", IJEDR, Volume 4, Issue 1, ISSN: 2321-9939.
- [12] Ohuru, Michael. (2019). Arduino as an ISP for AVR Microcontrollers (Atmega32).
- [13] Radek Fujdiak,, Jiří Mišurec, Petr Mlýnek, Ondřej Rášo, "Cryptography in Ultra-Low Power Microcontroller MSP430", International Journal of Engineering Trends and Technology (IJETT) – Volume 6 Number 8- Dec 2013.
- [14] <http://www.ti.com/lit/ds/symlink/msp430f5529.pdf>
- [15] <http://www.ti.com/lit/ug/slau533d/slau533d.pdf>
- [16] <http://www.kodenshi.co.jp/products/p-32.php>
- [17] <https://www.digikey.com/products/en?dc=21102&fid=804194>
- [18] <https://www.digikey.com/catalog/en/partgroup/bc417143/21102>
- [19] <https://cdn.sparkfun.com/datasheets/Wireless/Bluetooth/CSR-B417-datasheet.pdf>
- [20] <https://texasmicrocontroller.blogspot.com/2017/09/hc-05-interfacing-with-msp430f5529.html>
- [21] <http://www.eecs.ucf.edu/seniordesign/fa2016sp2017/g22/doc/ProjectDocumentation.pdf>
- [22] https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-2486-8-bit-AVR-microcontroller-ATmega8_L_datasheet.pdf