**Embedded Security Systems**

**Laboratory report for Lists 1 and 2**

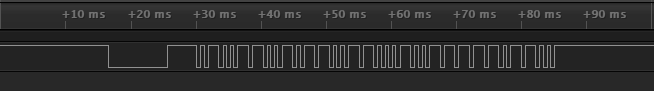
**Mikołaj Grzegrzółka 241073, Grzegorz Zaborowski 236447**

**List 1: IrDA and oscilloscope showcase:**

Technique and equipment used:

* Arduino Board Uno Rev 3
* IR Receiver Module TSOP31236 36 kHz
* IR Diode TSUS4300 3mm 950nm
* 1k Ω resistor
* Laptop with software
* Some wires and breadboard

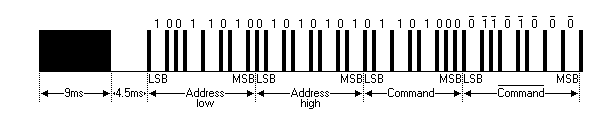
According to instruction we acquired IrDA trace from the email that was sent for decoding. As logic analyser we used Saleae and trace opened in it looked as shown in the picture below:



After checking few options and possibilities with other standard protocols used, we find out that in the trace sent for decoding the NEC protocol was used.

What is NEC protocol?

The Extended NEC Protocol message starts burst with a 9 ms delay followed by a cyclic interval of 4.5 ms. Then the address and the execution command are presented. The standard length of the address is 16 bits, and the command itself is sent again after the first transmission, i.e. 8 bits + 8 bits. All bits in this case are reversed a second time, with the possibility of using in the process of verifying the received message.

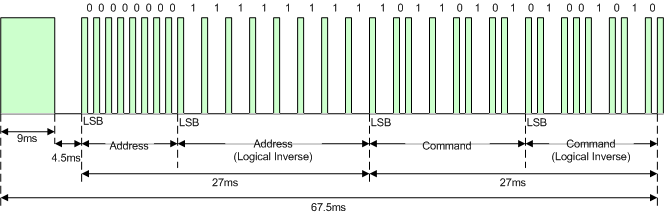


For the NEC protocol, bit coding based on pulse distance is used. The length of each pulse is 0.5625 ms at a carrier frequency of 38 KHz. The bit transmission itself looks like this:

Logical value 0: Burst of the pulse followed by a space of 0.5625 ms (sum of the total transmission time is 1.125 ms).

Logical value 1: Standard 562.5 µs pulse burst, but followed by a space of 3 x 0.5625 ms (so 1.6875 ms), with a total transmission time of 4 x 0.5625 ms (so 2.250 ms).

The actual matching pictures with our protocol and found picture of NEC protocol:



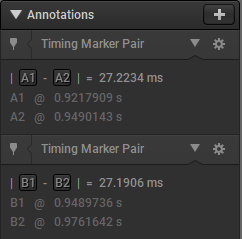
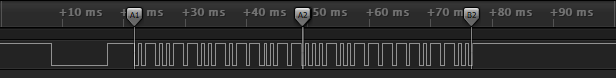
NEC message frame

* Leading pulse 9 ms
* Interval 4.5 ms
* 8-bit address
* 8-bit logical inverse of address
* 8-bit command
* 8-bit logical inverse of command

Signal fundamentals in our case looked like this:

Obraz zawierający tekst

Opis wygenerowany automatycznieObraz zawierający tekst

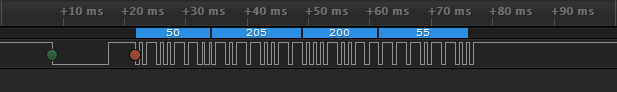
Opis wygenerowany automatycznie

So actually the message starts with 9 ms burst, then the burst is followed by space of 4.5 ms value. After that we have full message and command used. As it was already mentioned the value of logical 0 is 1.125 ms (sum of the total transmission) and the value of logical 1 is so 2.250 ms.

We searched for software to decode the NEC protocol and added the NEC protocol analyser for Saleae. Then the decoded information we got was:

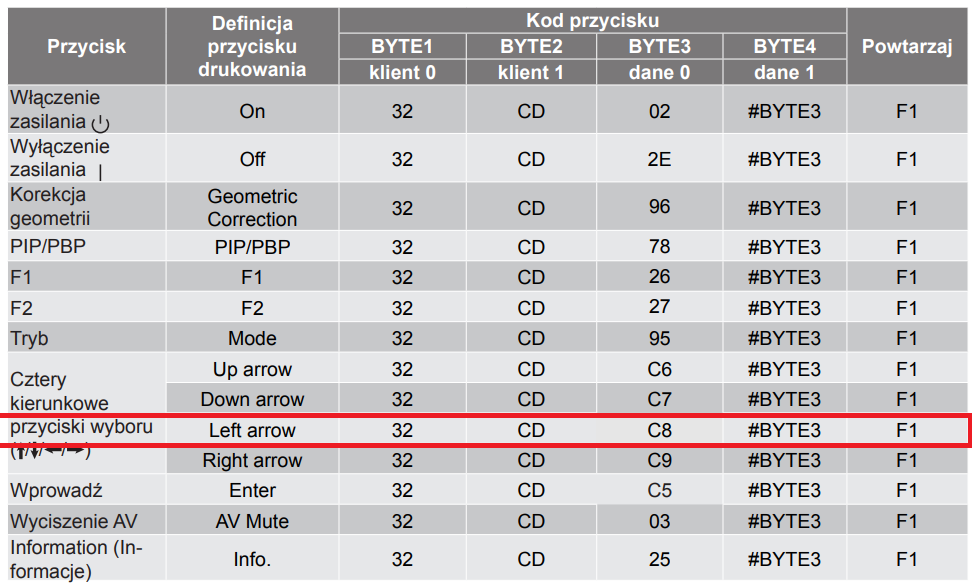
Obraz zawierający tekst

Opis wygenerowany automatycznieObraz zawierający tekst

Opis wygenerowany automatycznie

So we got: 0x32 0xCD 0xC8 0x37

We managed to establish that this is the control code for the OPTOMA remote controler button. Remote control used with DLP® projectors.

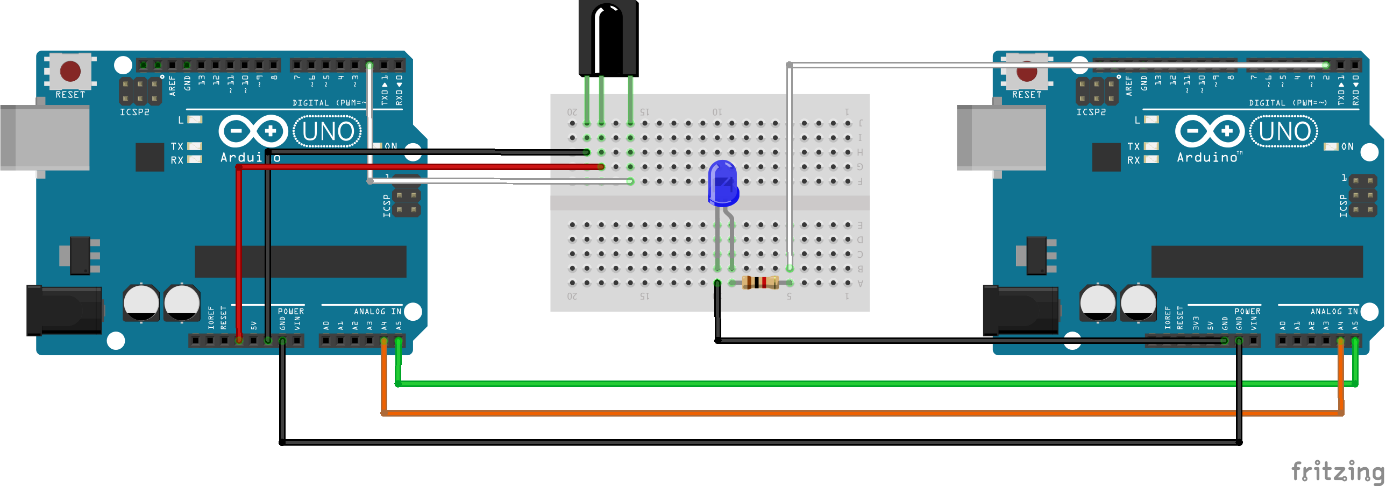


**List 2: Securing IrDA transmission with A5/1:**

Technique and equipment used:

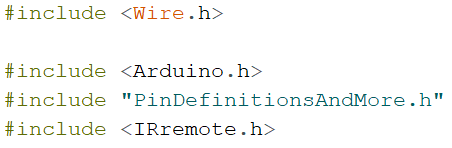
* Arduino Board Uno Rev3
* Arduino Board Clone Uno Rev3 ATMEGA328 CH340
* IR Receiver Module TSOP31236 36 kHz
* IR Diode TSUS4300 3mm 950nm
* 1k Ω resistor
* Laptop with software
* Some wires and breadboard

We connected Arduino boards with our laptops to communicate with them using UART. Using UART we send the message ‘Hello Word’ to first Arduino (Master). It generated random keystream for the message and transmit it to other Arduino (Slave) using separate communication channel which was I2C. Then the communication between Arduinos was established via IrDA link and the encrypted message was send.

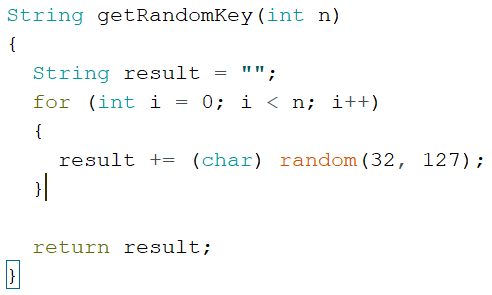


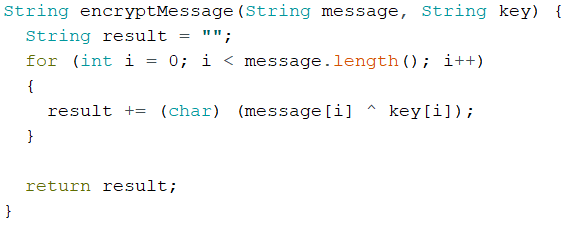
Wiring diagram

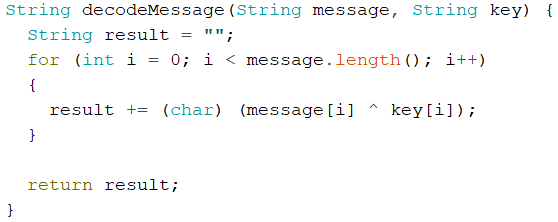
**Libraries:**



**Functions:**



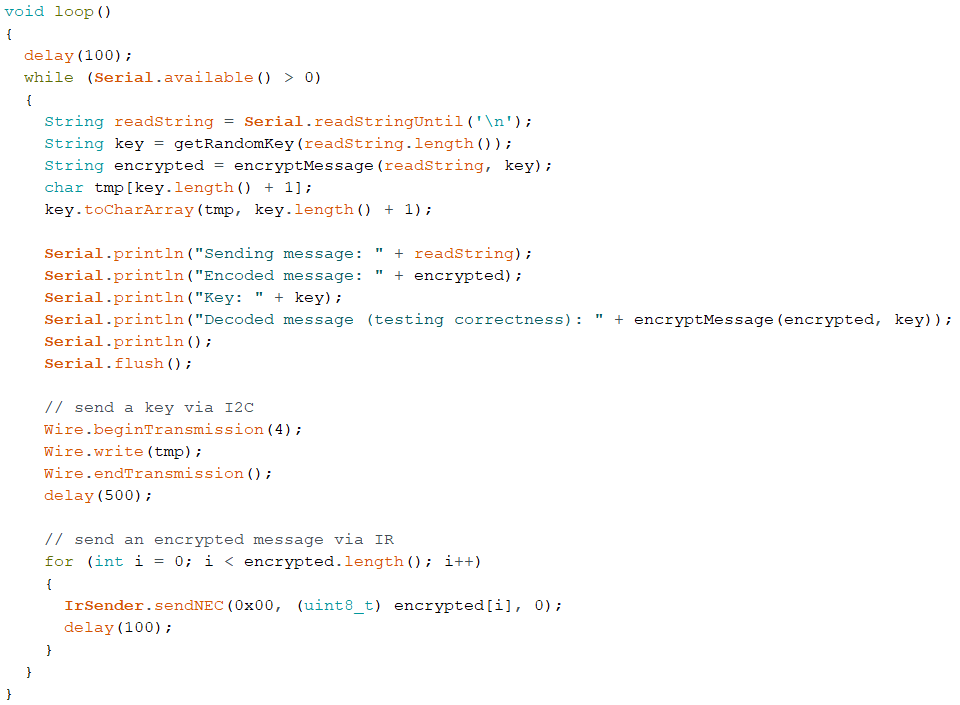




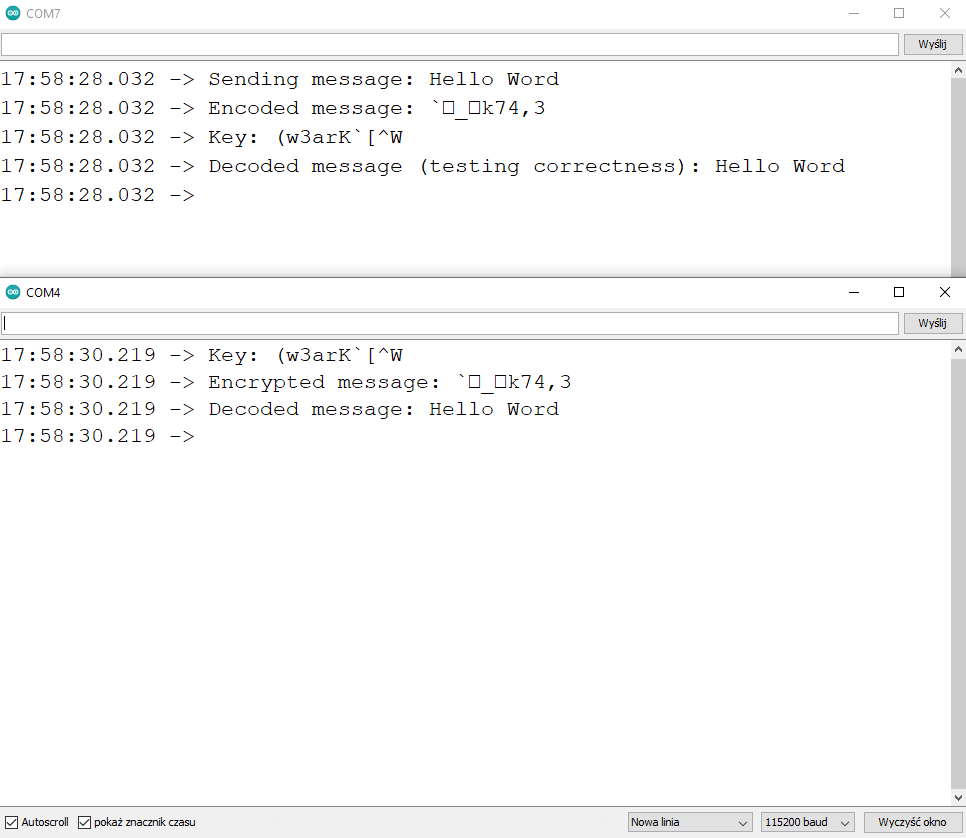
**Slave main loop:**

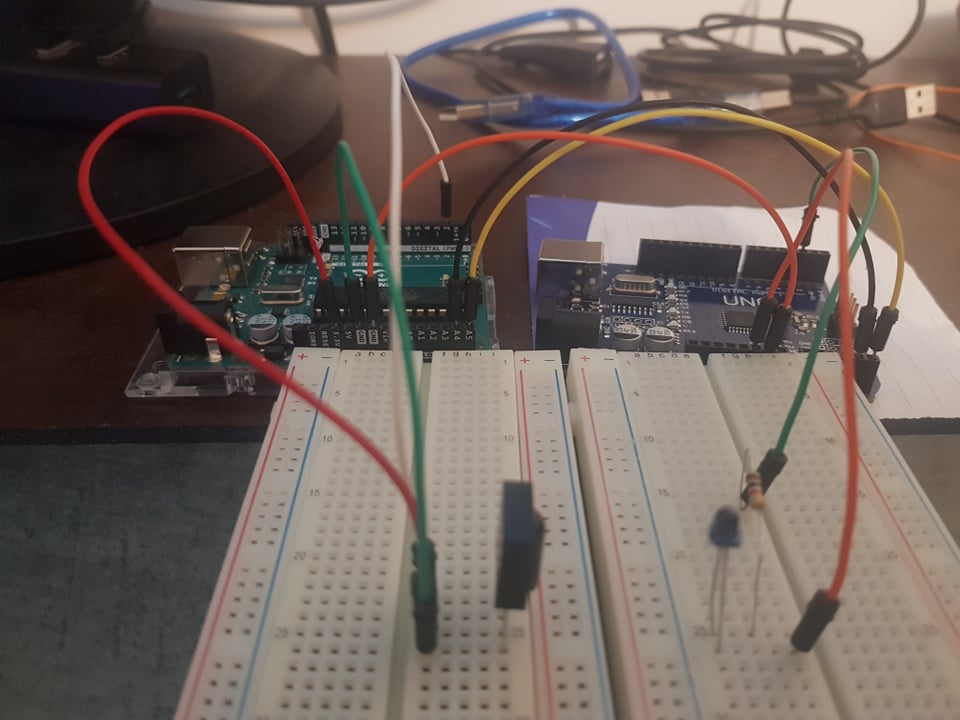


**Master main loop:**

****

**Result:**



****

**Conclusion:**

**We were able to successfully encode, transmit and decode the sent message. We also successfully used two different communication channels (IrDA and I2C). The solution is not optimized, as exemplified by the use of only one byte in the NEC protocol. There are serious issues to consider regarding code optimization and data transfer itself.**

**Bibliography:**

https://www.arduino.cc/en/Tutorial/LibraryExamples/MasterWriter

http://elonics.in/breadboard-projects/remote-tester-circuit-using-tsop1738

https://techdocs.altium.com/display/FPGA/NEC+Infrared+Transmission+Protocol

http://www.elenota.pl/datasheet-pdf/156574/Vishay/TSOP31230

https://docs.rs-online.com/1f00/0900766b80e22d5f.pdf