

Politechnika Warszawska

Systemy czasu rzeczywistego i sieci przemysłowe

Projekt nr 2

Wykonał:

Bartłomiej Guś gr. IPAUT-161

Warszawa 2020/2021

Spis treści

1. Wstęp.....	3
2. Odtworzyć kod dostępu do zamka składu broni.....	3
3. Ile jest możliwych kodów dostępu?	3
4. Jakie informacje należałoby zdobyć, aby kod został wyznaczony w sposób jednoznaczny?.....	3
5. Oświadczenie	6

1. Wstęp

Podczas tego ćwiczenia miałem możliwość zapoznania się z procedurą wyznaczenia kodu CRC a w szczególności wyznaczania tej procedury dla sieci MODBUS w trybie RTU.

2. Odtworzyć kod dostępu do zamka składu broni.

Pole adresu:	0x0A
Pole funkcji:	0x10
Adres pierwszego rejestru kodu:	0x0000
Adres ostatniego rejestru kodu:	0x0008
Liczba bajtów pola danych:	0x12
Pole danych:	0x1111
	0x1111
	0x1111
	0x2505
	0x1999
	0x1111
	0x1111
	0x1F38
CRC kodu:	0xA77C
CRC ramki:	0x9262

Tabela 1 - Ramka

3. Ile jest możliwych kodów dostępu?

W celu obliczenia CRC można posłużyć się poniższym wzorem:

$$W(x) \cdot x^n = W_G(x) \cdot I(x) + R(x)$$

gdzie:

$W(x)$ – przesyłana wiadomość

n – stopień wielomianu generującego, w tym przypadku wynosi: 16

$W_G(x)$ – wielomian generacyjny, w przypadku MODBUS RTU wynosi: $x^{16} + x^{15} + x^2 + 1$, co odpowiada $0xA001_h$

$I(x)$ – iloraz dzielenia

$R(x)$ – wartość reszty z dzielenia – wartość CRC

Ze względu na to, że kod posiada 128 bitów, a kod CRC jest 16 bitowy co świadczy o tym, że musimy znać 112 bitów kodu, aby wyznaczyć jednoznacznie pozostałe 16 bitów. Z tego wynika, że w przypadku kodu o długości 128 bitów istnieje 2^{112} różnych kombinacji, czyli prawdopodobieństwo zgadnięcia kodu jest niewiarygodnie małe wynosi ok. $2 \cdot 10^{-34}$. W przypadku gdyby kod ten miał długość jedynie 16 bitów i znalibyśmy kod CRC (również 16 bitowy) istniałaby wtedy jedynie jedna kombinacja bitów kodu spełniająca wyliczoną sumę kontrolną.

4. Jakie informacje należałoby zdobyć, aby kod został wyznaczony w sposób jednoznaczny?

W celu jednoznacznego wyznaczenia kodu o długości 128 bitów należałoby dowiedzieć się co najmniej 112 bitów z kodu, aby jednoznacznie określić jego całkowitą postać.

5. Kod źródłowy programu za pośrednictwem, którego obliczyłem wartości pola danych

```
#include <iostream>
#include <bitset>

using namespace std;

int licznik = 0;
unsigned char p[16];
void Sprawdzanie(int ktory_wyraz);

unsigned short CRC(unsigned char *pMessage, unsigned int NumberOfBytes)
{
    register unsigned short reg16 = 0xFFFF;
    unsigned char reg8;
    unsigned char i;

    while (NumberOfBytes--)
    {
        reg16 ^= *pMessage++;
        i = 8;

        while(i--)
        {
            if (reg16 & 0x0001)
            {
                reg16 >>= 1;
                reg16 ^= 0xA001;
            }
            else
                reg16 >>= 1;
        }
    };

    reg8 = reg16 >> 8;

    return (reg16);
}

void Sprawdzanie(int ktory_wyraz)
{
    int wyraz_nastepny = ktory_wyraz+1;

    for(int i = 0; i<256; i++)
    {
        p[ktory_wyraz] = {i};

        if(ktory_wyraz<15)
        {
            Sprawdzanie(wyraz_nastepny);
        }
        else if (ktory_wyraz==15)
        {
            unsigned short wartosc = CRC(p,16);

            bitset<16> bitset1(wartosc);

            bitset<16> bitset2({0x7CA7});

            if(bitset1==bitset2)
            {
                licznik++;
                cout<<licznik<<endl;
            }
        }
    }
}
```

```

int main()
{
    for(int i = 0; i<16;i++)
    {
        p[i]={0x11};
    }

    //    Sprawdzanie(0); // Funkcja służąca do sprawdzenia liczby rozwiązań
    //    cout<<licznik;

    p[6]={0x25};
    p[7]={0x05};
    p[8]={0x19};
    p[9]={0x99};

    p[14] = {0x00};
    p[15] = {0x00};

    bool czy_odbryl = false;

    int licznik = 0;

    for(int i = 0; i<256;i++)
    {
        p[14]={i};

        licznik = 0;

        do{

            unsigned short wartosc = CRC(p,16);

            bitset<16> bitset1(wartosc);

            bitset<16> bitset2({0x7CA7});

            if(bitset1==bitset2)
            {
                czy_odbryl = true;

                cout<<bitset1<<endl;
                cout<<bitset2<<endl;

                bitset<8>bitset3(p[15]);

                cout<<"Ostatni:"<<endl;
                cout<<bitset3<<endl;
            }

            p[15]++;

            licznik++;

        }while(!czy_odbryl&&licznik<256);

        if(czy_odbryl)
        {
            break;
        }
    }

    bitset<8>bitset3(p[14]);

    cout<<bitset3<<endl;

    return 0;
}

```

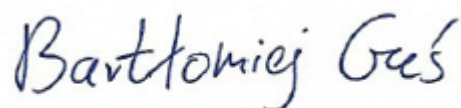
6. Oświadczenie

Warszawa, 16.05.2021r.

Oświadczenie

Oświadczam, że niniejsza praca stanowiąca podstawę do uznania osiągnięcia efektów uczenia się z przedmiotu „Systemy czasu rzeczywistego i sieci przemysłowe” została przeze mnie wykonana samodzielnie.

Bartłomiej Guś nr albumu 297415

Handwritten signature of Bartłomiej Guś in blue ink.