

Implementacja komunikacji
pomiędzy wirtualnym terminalem a
mikrokontrolerem MSP430 oraz
pomiędzy dwoma mikrokontrolerami
MSP430 z wykorzystaniem kodu
Morse'a.
Projekt wprowadzający.

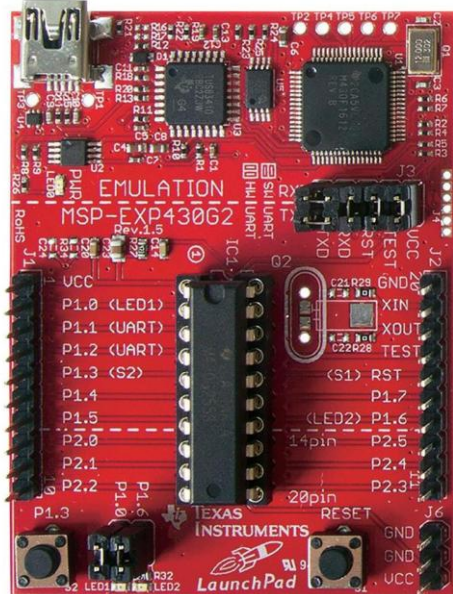
Krzysztof Olej
AiR sem. 5. 2016/2017
30.01.2017

Założenia projektu:

- Kodowanie ciągu znaków wpisanego w wirtualny terminal na kod Morse'a
- Dekodowanie kodu Morse'a i wyświetlanie ich na wirtualnym terminalu
- Komunikacja pomiędzy dwoma mikrokontrolerami za pomocą kodu Morse'a

Projekt wykonano przy użyciu mikrokontrolera MSP-EXP430G2 oraz płytki uruchomieniowej LaunchPad firmy Texas Instruments.

LaunchPad MSP430:



Środowisko wykorzystane do napisania kodu oraz zaprogramowania mikrokontrolera to Energia – open-sourcowe, zmodyfikowane środowisko programistyczne Arduino IDE.

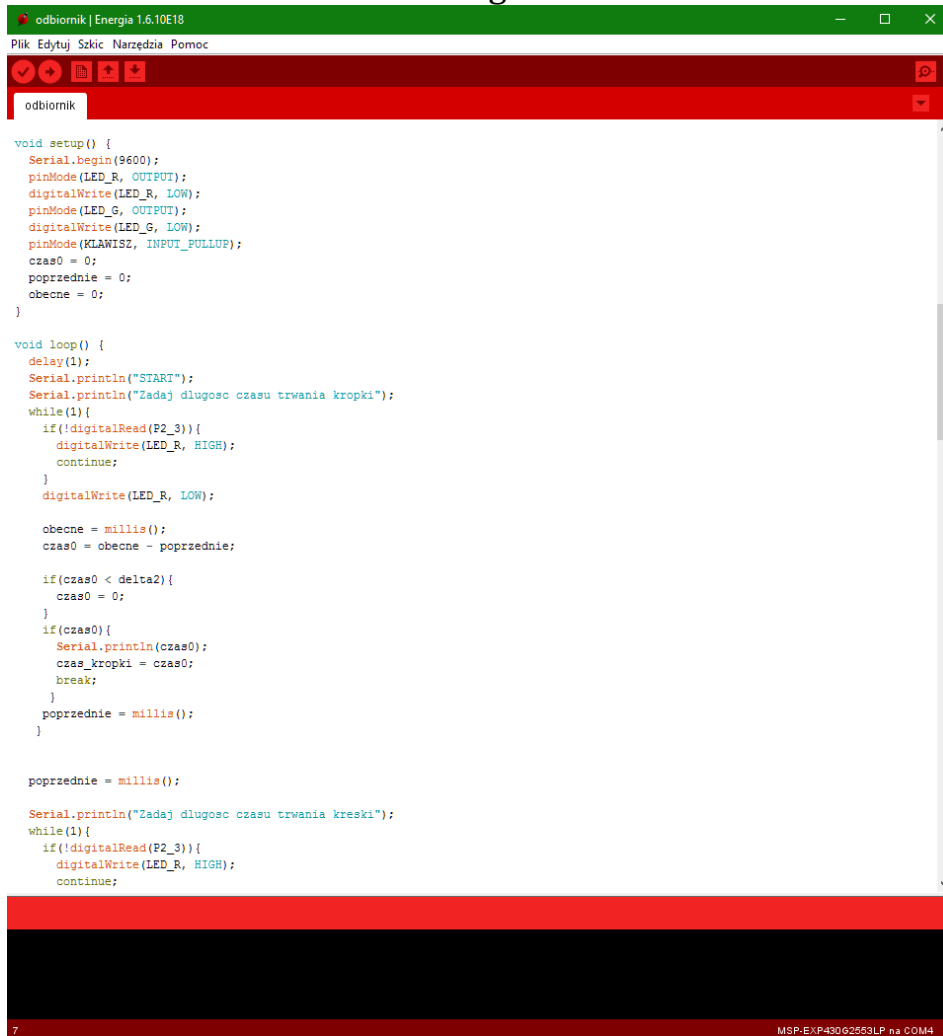


Energia^{MT}

Energia^{MT} is a modified version of the Arduino/
Genuino IDE for the Texas Instruments MSP430,
MSP432, Tiva C and CC3200 LaunchPad boards.

This software is not supported by the Arduino LLC.

Zrzut ekranu środowiska Energia:



```
odbiornik | Energia 1.6.10E18
Plik Edytuj Szukaj Narzędzia Pomoc
odbiornik

void setup() {
  Serial.begin(9600);
  pinMode(LED_R, OUTPUT);
  digitalWrite(LED_R, LOW);
  pinMode(LED_G, OUTPUT);
  digitalWrite(LED_G, LOW);
  pinMode(KLAWISZ, INPUT_PULLUP);
  czas0 = 0;
  poprzednie = 0;
  obecne = 0;
}

void loop() {
  delay(1);
  Serial.println("START");
  Serial.println("Zadaj dlugosc czasu trwania kropki");
  while(1){
    if(!digitalRead(P2_3)){
      digitalWrite(LED_R, HIGH);
      continue;
    }
    digitalWrite(LED_R, LOW);

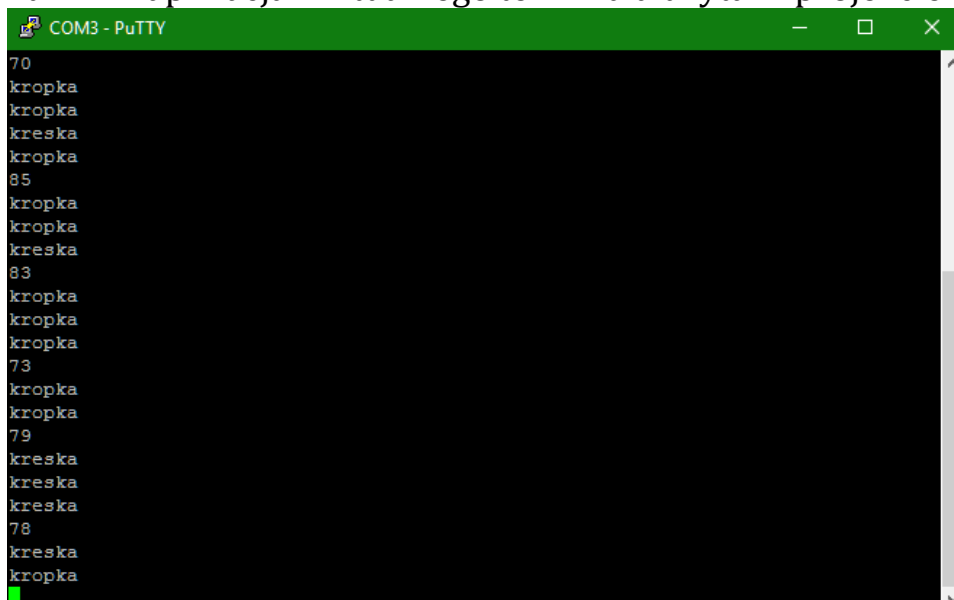
    obecne = millis();
    czas0 = obecne - poprzednie;

    if(czas0 < delta2){
      czas0 = 0;
    }
    if(czas0){
      Serial.println(czas0);
      czas_kropki = czas0;
      break;
    }
    poprzednie = millis();
  }

  poprzednie = millis();
  Serial.println("Zadaj dlugosc czasu trwania kreski");
  while(1){
    if(!digitalRead(P2_3)){
      digitalWrite(LED_R, HIGH);
      continue;
    }
  }
}
```

MSP-EXP430G2553LP na COM4

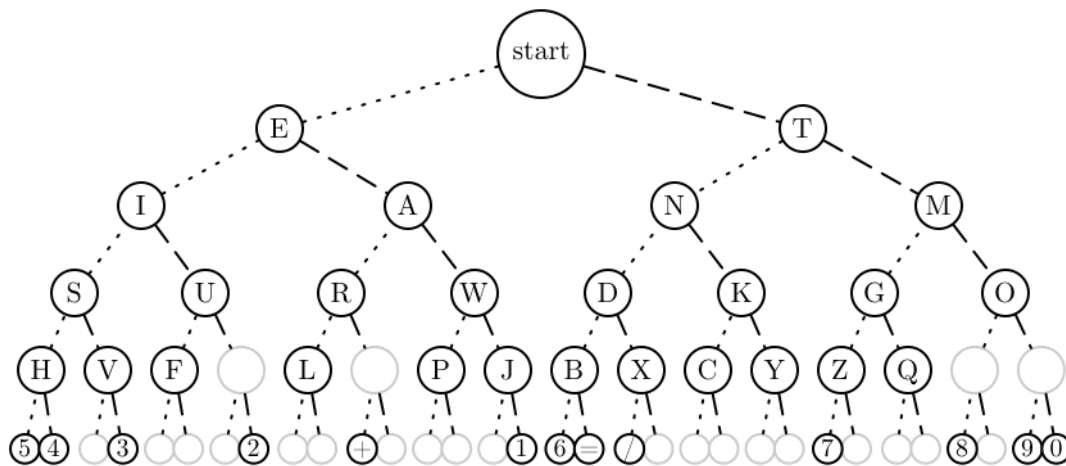
PuTTY – aplikacja wirtualnego terminala użyta w projekcie:



```
COM3 - PuTTY
70
kropka
kropka
kreska
kropka
85
kropka
kropka
kreska
83
kropka
kropka
kropka
73
kropka
kropka
79
kreska
kreska
kreska
78
kreska
kropka
```

Kodowanie ciągu znaków wpisanego w wirtualny terminal na kod Morse'a.

Mikrokontroler przy użyciu portu szeregowego odbiera znaki w kodzie ASCII – litery oraz cyfry. Następnie mikrokontroler wyświetla zakodowane znaki za pomocą zielonej i czerwonej diody wbudowanej w płytke – odpowiednio kropka i kreska. Zdefiniowany czas świecenia zielonej diody to 500 milisekund, czerwonej 1500. Implementacja kodowania znaku wykorzystuje graf Morse'a oraz rekurencję.



Dekodowanie kodu Morse'a i wyświetlanie ich na wirtualnym terminalu.

Użytkownik wprowadza znak kodem Morse'a przy użyciu wbudowanego przycisku. Po uruchomieniu mikrokontrolera należy ustawić długość czasu trwania kropki oraz kreski. Długość czasu trwania wciśnięcia przycisku może różnić się o ± 200 milisekund w porównaniu do ustawionej wartości. Po wprowadzeniu ostatniej kropki lub kreski należy przytrzymać przycisk dłużej niż *czas trwania kreski + 200 milisekund*, spowoduje to dopasowanie wprowadzonego ciągu kropek i kreszek do znaku, a następnie wyświetlenie na wirtualnym terminalu odkodowanej litery lub cyfry. W przypadku wprowadzenia błędnego ciągu kropek i kreszek zostanie wyświetlony komunikat: „bład znaku”.

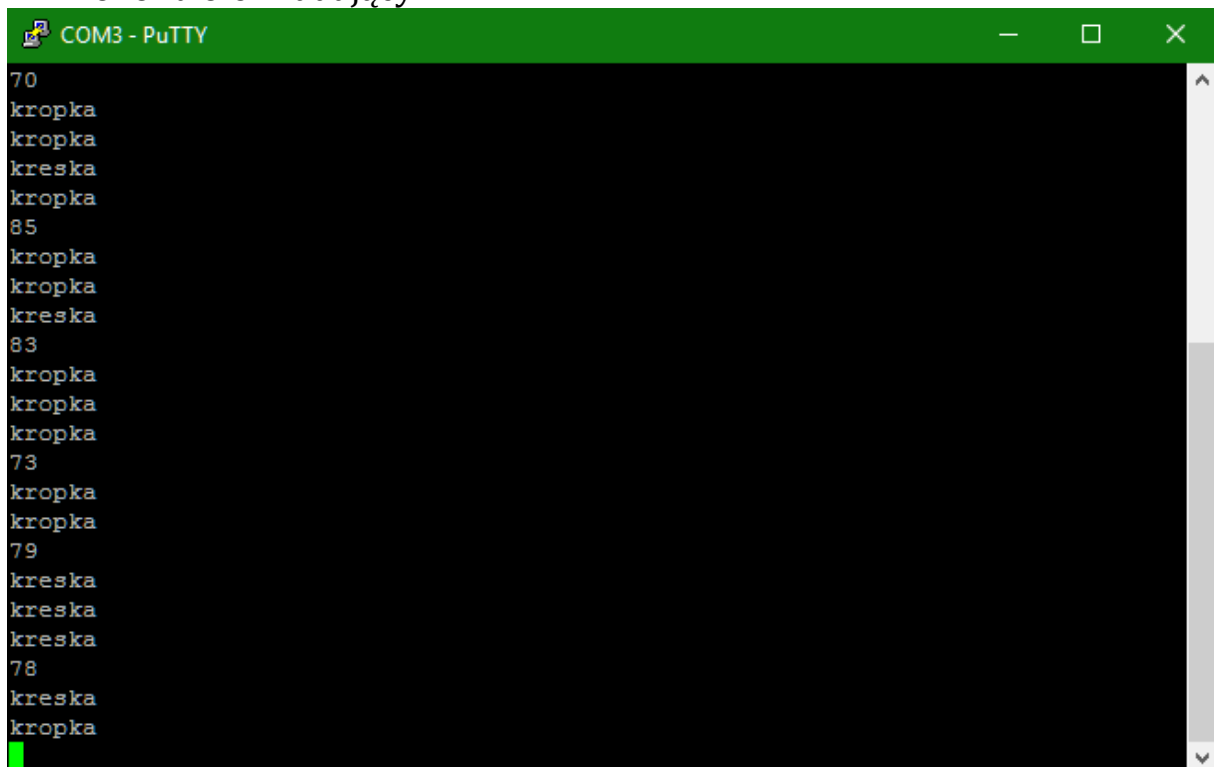
Komunikacja między dwoma mikrokontrolerami za pomocą kodu Morse'a

Mikrokontroler nadający kod Morse'a przed rozpoczęciem wysyłania ustawia długość czasu trwania kropki i kreski – mikrokontroler odbierający wymaga ustawienia czasu po każdym włączeniu, dzięki temu może on odbierać impulsy o różnych prędkościach bez konieczności przeprogramowywania. Mikrokontroler nadający wysyła impulsy do drugiego mikrokontrolera – kropki (krótsze impulsy), kreski (dłuższe impulsy) albo impuls końca znaku.

Mikrokontroler odbierający interpretuje impulsy, a następnie wysyła odkodowane znaki przez port szeregowy. Schemat działania mikrokontrolerów odbierającego i nadającego jest bardzo podobny do schematu działania w poprzednich etapach projektu. Mikrokontrolery zostały połączone za pomocą transoptora EL817B w celu izolacji.

Zrzuty ekranu z programu wirtualnego terminala:

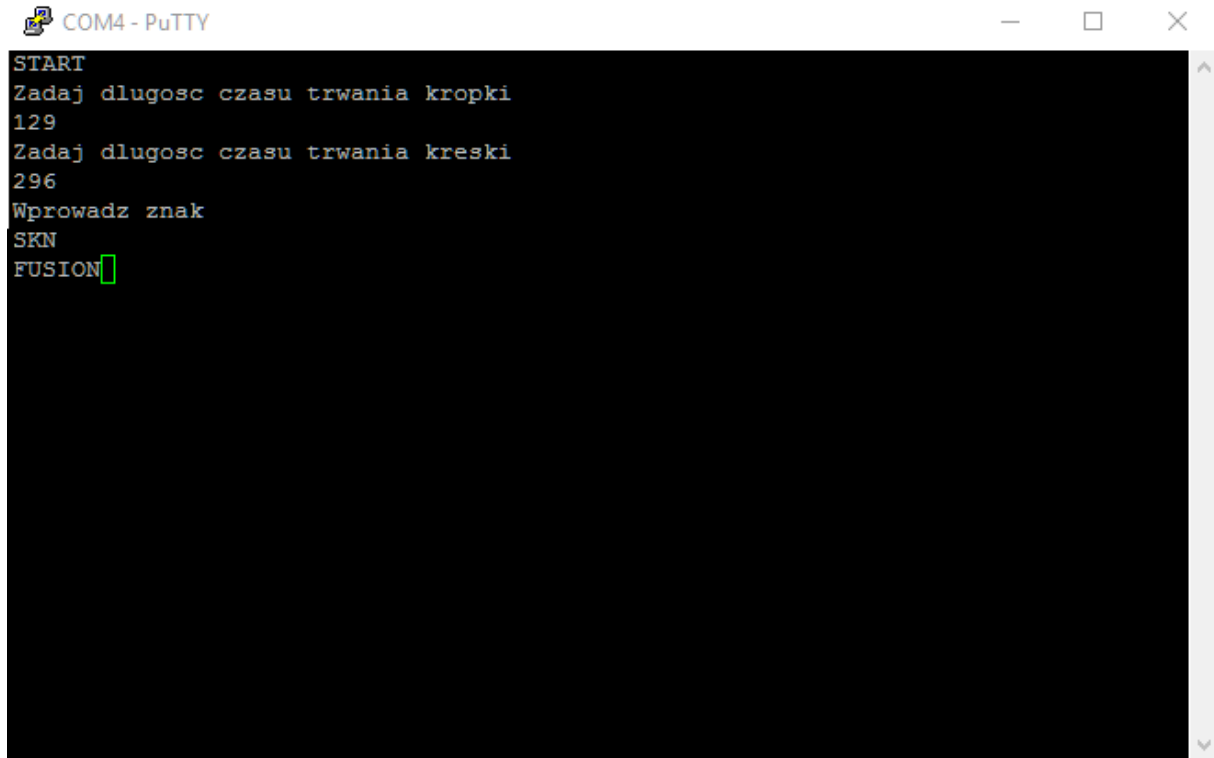
-mikrokontroler nadający:



```
COM3 - PuTTY
70
kropka
kropka
kreska
kropka
85
kropka
kropka
kreska
83
kropka
kropka
kropka
73
kropka
kropka
79
kreska
kreska
kreska
78
kreska
kropka
```

W aplikacji wirtualnego terminala wypisywany jest kod ASCII kodowanego znaku a następnie ciąg napisów „kropka” lub „kreska” w zależności od tego jaki kod ma dany znak.

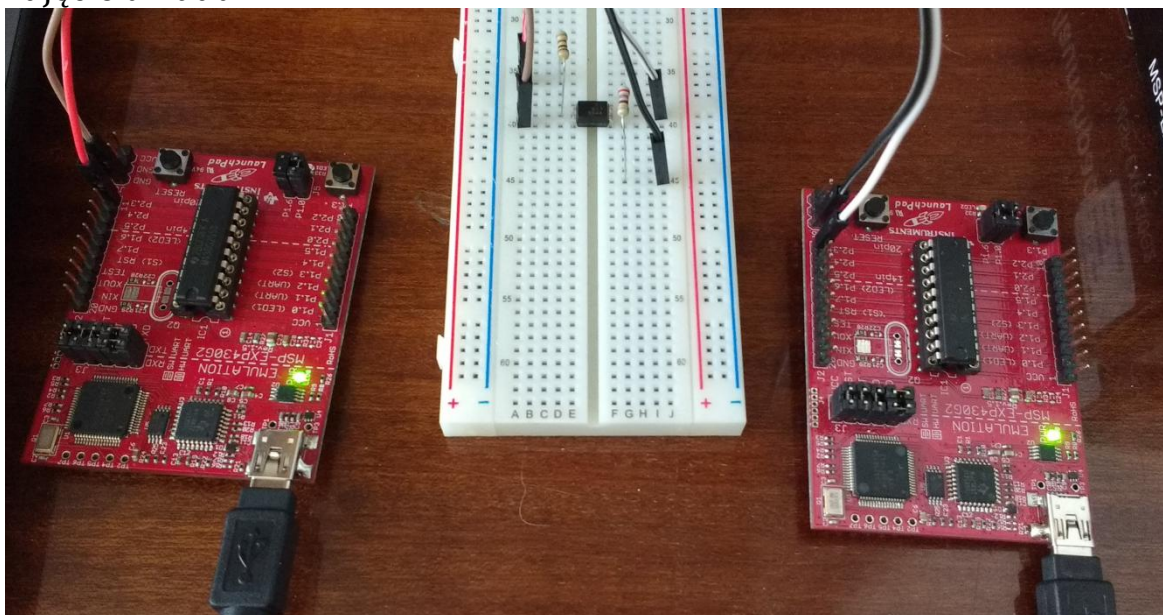
-mikrokontroler odbierający:



```
COM4 - PuTTY
START
Zadaj dlugosc czasu trwania kropki
129
Zadaj dlugosc czasu trwania kreski
296
Wprowadz znak
SKN
FUSION
```

W aplikacji wirtualnego terminala wypisywane są odebrane znaki. Po napotkaniu znaku końca wyrazu kursor przechodzi do nowej linii.

Zdjęcie układu:



Elementy użyte w projekcie:

- LaunchPad MSP430G2 firmy Texas Instrument – 2 sztuki
- transoptor EL817B
- rezystor 260Ω
- rezystor 100Ω

Źródła:

Energia:

<http://energia.nu/download/>

Tutoriale dla MSP430G2:

[http://energia.nu/Guide MSP430LaunchPad.html](http://energia.nu/Guide_MSP430LaunchPad.html)

Przykłady w środowisku Energia

Wirtualny terminal PuTTY:

<http://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>