Systemy Wbudowane AGH  
Dokumentacja ogólna

# projekcja zegarka przy pomocy neopixeli na wahdle. Marek Fudaliński Marcin sośniak

1) Materiały:

* Arduino Pro Mini (dalej zwane arduino)
* Cewka do indukcyjnego zasilania Arduino pro Mini
* Zasilacz do zasilania indukcyjnego
* Neopixele (w zależności od chęci od 6 do 12 sztuk)
* Silnik elektryczny bez szczotkowy D1811-2000
* Zasilacz do silnika wraz z przyciskiem (HOME MADE)
* Czujnika Hala
* Magnes
* Podstawa
* Programator AVR Dragon (dalej zwany dragon)

2) Wykonanie:

* Część fizyczna:

1. Do podstawy należy przymocować silnik.
2. Silnik należy podpiąć do zasilacza.
3. Jedną część cewki umieścić nieruchomo względem podstawy połączyć z zasilaczem indukcyjnym.
4. Na godzinie 12 przymocuj magnes tak aby znajdował się możliwie blisko czujnika hala
5. Skonstruować WYWAŻONE wahadło w którym:
   * na jednym końcu znajduje się Arduino wraz z przeciwwagą.
   * na drugim znajdują się od góry Neopixele wraz z czujnikiem Hala od spodu.
   * zamocować nieruchomo cewkę względem wahadła tak aby jej rzut. pokrywał się z cewką na podstawie oraz znajdowała się możliwie blisko.
   * Wyjście RX1 na arduino podpiąć do DIN na listwie z neopixelami .
   * Wyjście 5V podpiąć do 5 V na listwie z neopixelami .
   * Wyjście GND podpiąć do GND na listwie z neopixelami .
   * GND hallotronu spiąć z GND na listwie z neopixelami.
   * 5V hallotronu spiąć z 5V na listwie z neopixelami.
   * Q hallotronu spiąć z TX0 na arduino.
6. Przymocować tak skonstruowane wahadło do silnika.
7. Zaprogramować arduino przy pomocy dragona
8. Zaprogramować zasilacz.
9. Podpiąć oba do zasilania
10. Uruchomić silnik przy pomocy przycisku

* Część programowa:

- Do realizacji zadania użyliśmy języka C++ oraz asemblera.  
- Przy obsłudze wyświetlania wykorzystaliśmy język C++ oraz assemblera.  
- Do obsługi silnika wykorzystaliśmy assembler.

* Szczegóły części programowej:

1. Pseudokod działania programu na arduino:
   * Zainicjalizuj wszystkie możliwe linie do wyświetlenia.
   * Zainicjalizuj timer aktualizujący godzinę.
     + Ustal rozdzielczość
     + Ustal przerwania
   * Zainicjalizuj timer wywołujący wyświetlenie linii
     + Ustal rozdzielczość
     + Ustal przerwanie
   * Zainicjalizuj startowa godzinę
   * while(True):
     + oczekuj na synchronizacje z godzina 12

(używając czujnika hala oraz magnesu)

* + - ustaw gotować do wyświetlenia klatki

1. Wyświetlanie linii jest wykonywane w przerwaniach i działa następująco:
   * Timer1 (16bit) ustawiony na:
     + Prescaler 1
     + Liczy do 8888
   * To przerwanie jest DŁUGIE ponieważ obsługuje całość IO (~530 μs)
   * Wyświetl aktualną linię.
   * Zaktualizuj numer linii do wyświetlenia
   * Wylicz następująco linię do wyświetlenia
2. Aktualizowanie godziny jest wykonywane w przerwaniu i działa następująco:
   * Timer ustawiony następująco
     + Prescaler 1024
     + Liczy do 125
   * To przerwanie jest krótkie jednak musi odbywać się rządniej niż raz na 530 μs z uwagi na brak obsługi zagnieżdżonych przerwań.
   * Przerwanie aktualizuje licznik modulo 125 gdy zliczy do 0 aktualizuje godzinę (dodaje jedna sekundę do aktualnej godziny)
3. Modyfikacja stałych globalnych pozwala na:
   * Zmianę koloru wskazówek (najlepszy efekt przy najjaśniej ustawionych kolorach)
   * Zmianę wielkości wskazówek
   * Zmianę początkowej godziny

3) Możliwe poprawki:

* Dodanie RTC aby wyświetlać prawdziwa godzinę (raczej dość małe, żeby wybalansować ramie)
* Możliwe jest tez znaczne zwiększenie ilości diod poprzez zrównoleglenie i inny sposób wysyłania (do 12\*8=96) nie zwiększając potrzebnej ilości pamięci na każda diodę (uważać na ilość pamięci, ale można tez w segmencie kodu powstawać prewyliczone ramki, i zyskać tym kilka KB pamięci)
* pojawia się tez problem z zasilaniem, trzeba zmodyfikować zasilanie indukcyjne (więcej mocy)
* Rozważyć przesyłanie danych bez przewodowo, ale możliwe zbyt duże zakłócenia od zasilacza indukcyjne