Elektronika w urządzeniach medycznych

Projekt

Temat:

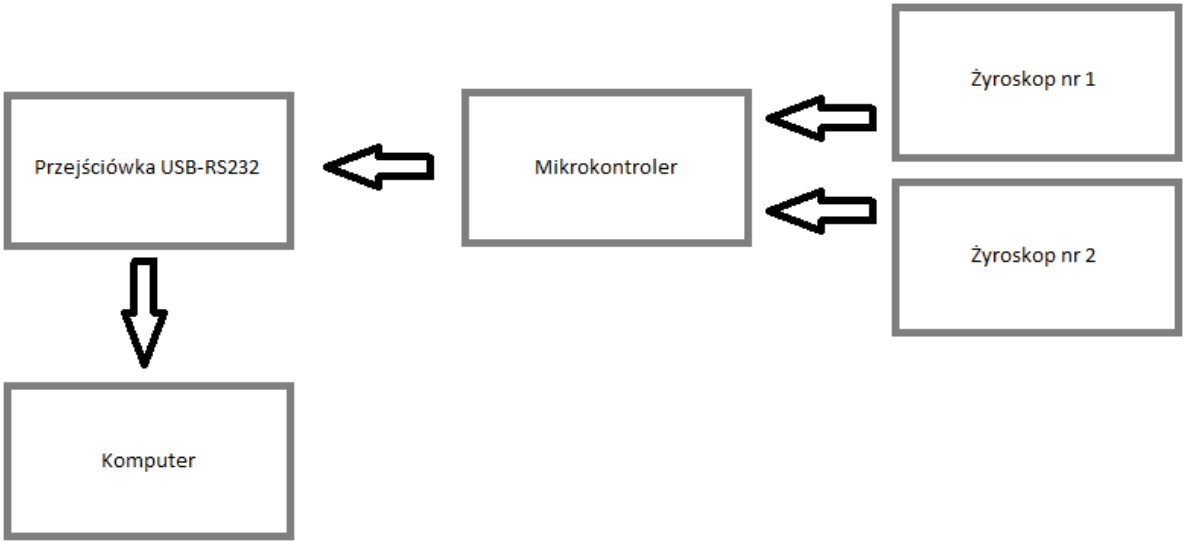
Urządzenie do pomiaru zgięcia stawu kolanowego

Wykonawca: Marek Piotrowski

Prowadzący:

Mgr inż. Roman REGULSKI

1. Schemat ogólny urządzenia



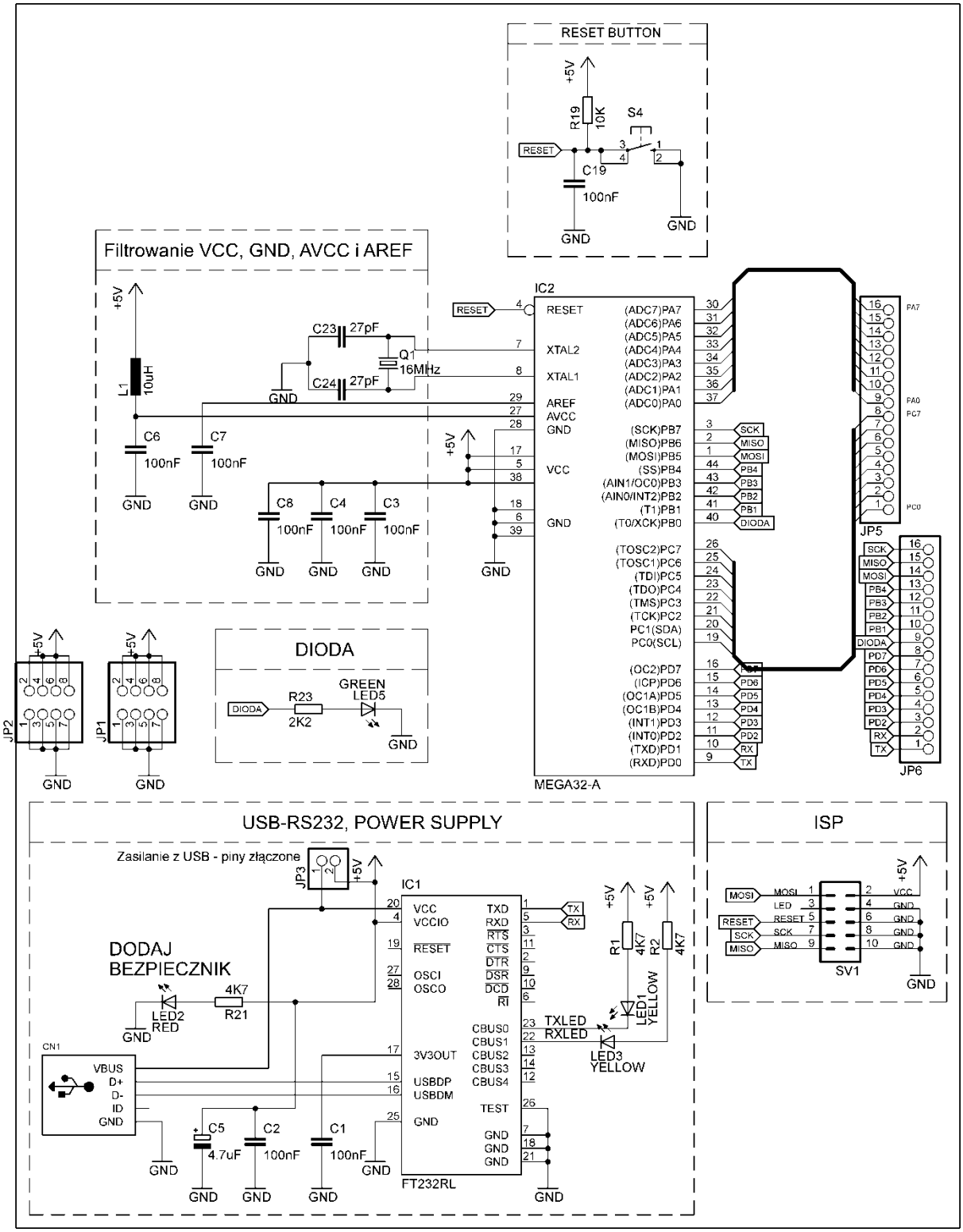
Rysunek 1 Schemat ogólny urządzenia

Mikrokontroler po inicjalizacji żyroskopów odczytuje z nich dane z częstotliwością 20Hz. Następnie wysyła je po wirtualnym RS232 do komputera i tam dane obsługuje program wizualizujący.

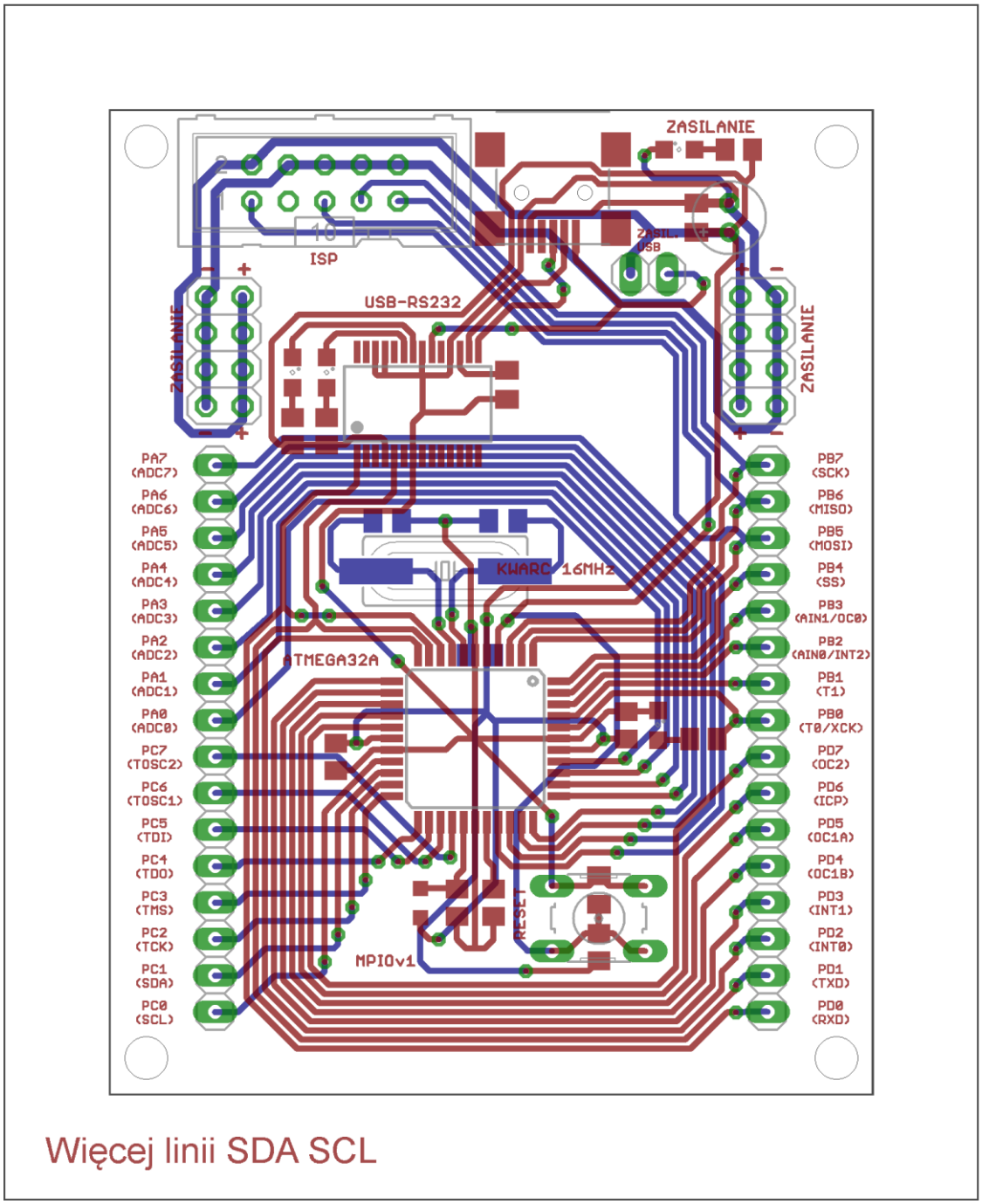
Układ można podzielić na dwa główne elementy:

a) płytka prototypowa z przejściówką USB-RS232,
b) żyroskopy.

2. Płytka prototypowa



Schemat 1 Płytka prototypowa



Schemat 2 Rozmieszczenie ścieżek płytki prototypowej

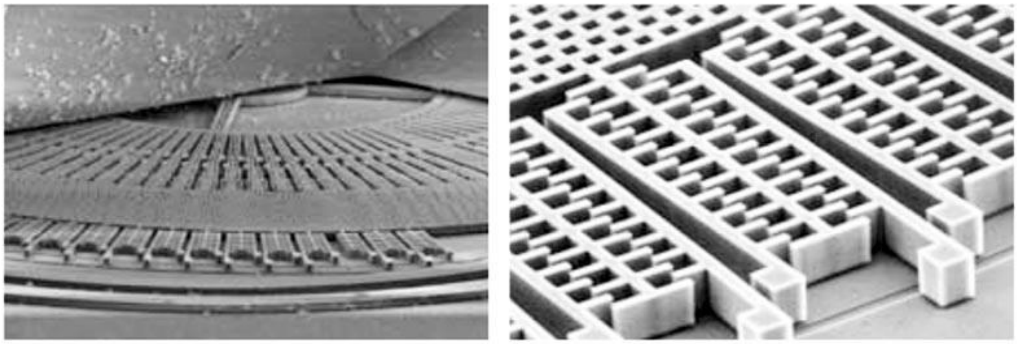
Założenia konstrukcyjne płytki prototypowej:
- wyprowadzenie wszystkich nóżek mikrokontrolera na goldpiny,
- przycisk resetu mikrokontrolera,
- łatwa komunikacja z komputerem,
- dioda wykorzystywana do analizy działania programu,
- dioda sygnalizująca zasilanie układu,
- programowanie mikrokontrolera przez ISP.

3. Żyroskopy

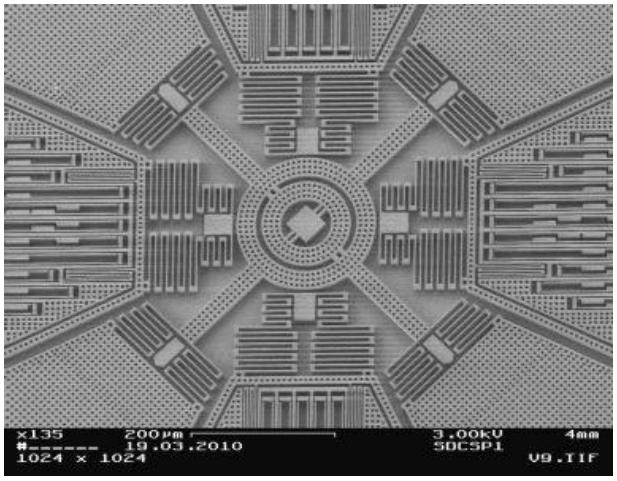
Pomiar kąta został zrealizowany na podstawie żyroskopów MPU-6050 wykonanych w technologii MEMS (mikrosystemów elektromechanicznych).

Rysunek 2. MPU-6050 [4]

Przykłady budowy mikrosystemów elektromechanicznych:



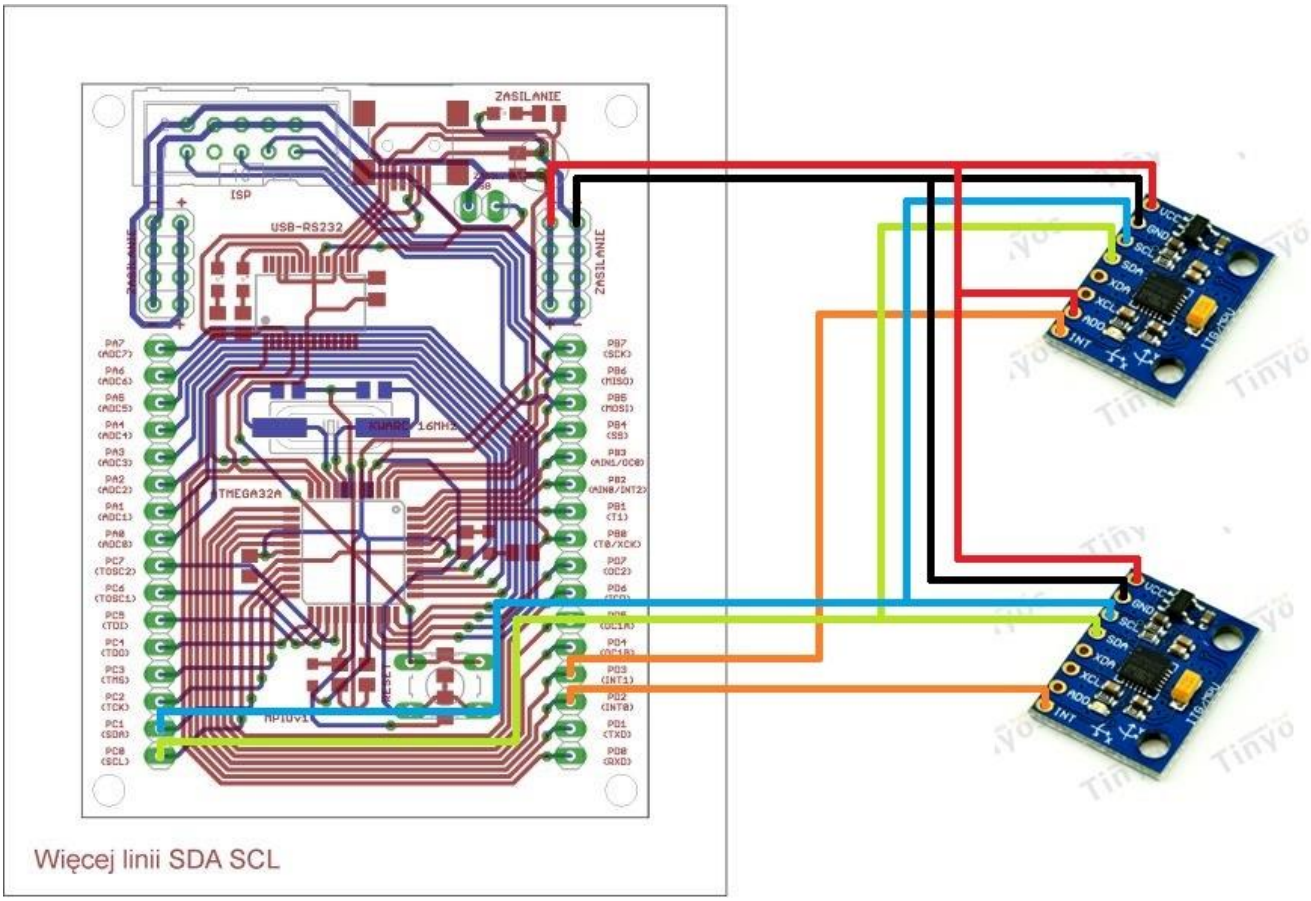
Rysunek 1. MEMS źródło: **www.mikrokontroler.pl**



Rysunek 2. Układ MEMS źródło: **www.sbanalyzer.pl**

Rysunek 3 Opis żyroskopu MPU6050

Zasilamy żyroskop 5V. Na płytce znajduje się stabilizator 3.3V, który zapewnia stabilne zasilanie chipowi. Nie zaobserwowałem problemów z dopasowaniem napięć na liniach komunikacyjnych.



Rysunek 4 Schemat połączenia układu

Żyroskopy łączymy z płytką prototypową pinami o identycznych nazwach: VCC, GND, SCL, SDA.

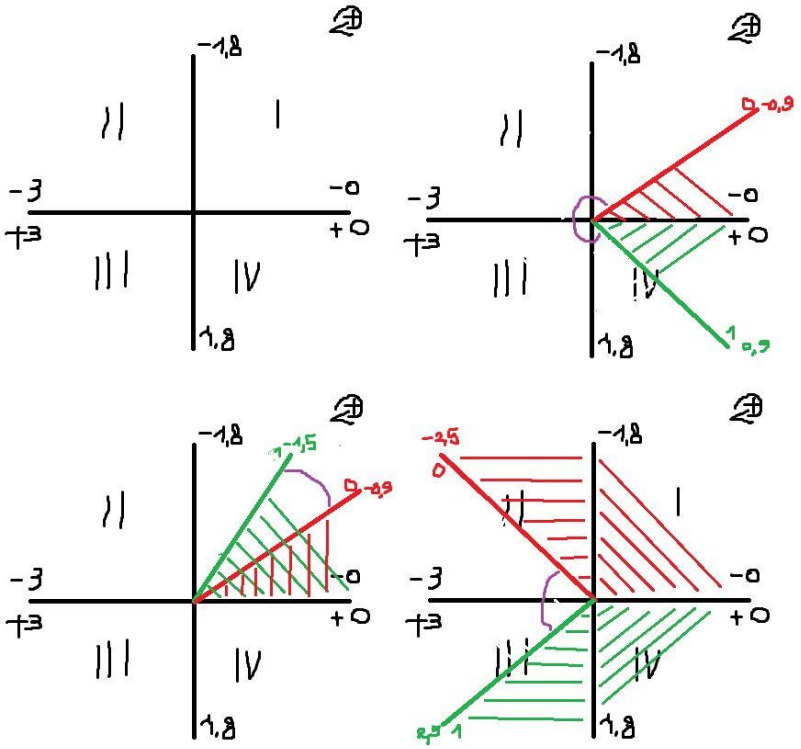
Różnice występują przy połączeniu INT i AD0. Żyroskop nr 1 połączony jest z INT0 i nie ma podciągnięcia do AD0. Żyroskop nr 2 połączony jest z INT1 i ma podciągnięcie AD0 do VCC.

4. Programowanie

Układ został zaprogramowany w Eclipse. Do oprogramowania żyroskopów wykorzystałem bibliotekę autorstwa Davide Gironi napisaną w 2012. Aby wykorzystać dwa żyroskopy musiałem ją zmodyfikować, a także zmienić działanie jednej z funkcji niższego rzędu obsługującą kolejkę danych z żyroskopu, ponieważ zatrzymywała działanie programu czekając na odpowiednią ilość bajtów pakietu danych.

Ze względu na dużą ilość obliczeń by użyć dwa żyroskopy trzeba było wykorzystać wbudowane w żyroskopy procesory DMP (Digital Motion Processor), które odciążyły procesor z filtracji danych.

Obliczając kąt pomiędzy żyroskopami należało uwzględnić ich układ współrzędnych



Rysunek 5 Obliczanie kąta pomiędzy żyroskopami

Funkcja obliczająca kąt pomiędzy żyroskopami:

```
float obliczKat(float a, float b)
{
  float kat;
  int polowkaA;
  int polowkaB;

  if(a<=0) polowkaA = 0;
  else polowkaA = 1;

  if(b<=0) polowkaB = 0;
```

```
  else polowkaB = 1;

  if(a<0) a=-a;
  if(b<0) b=-b;

  polowkaA += polowkaB;

  if(polowkaA ==0 || polowkaA ==2) kat = a-b;
  else kat = a+b;
  if(kat<0) kat=-kat;

  return kat;
}
```

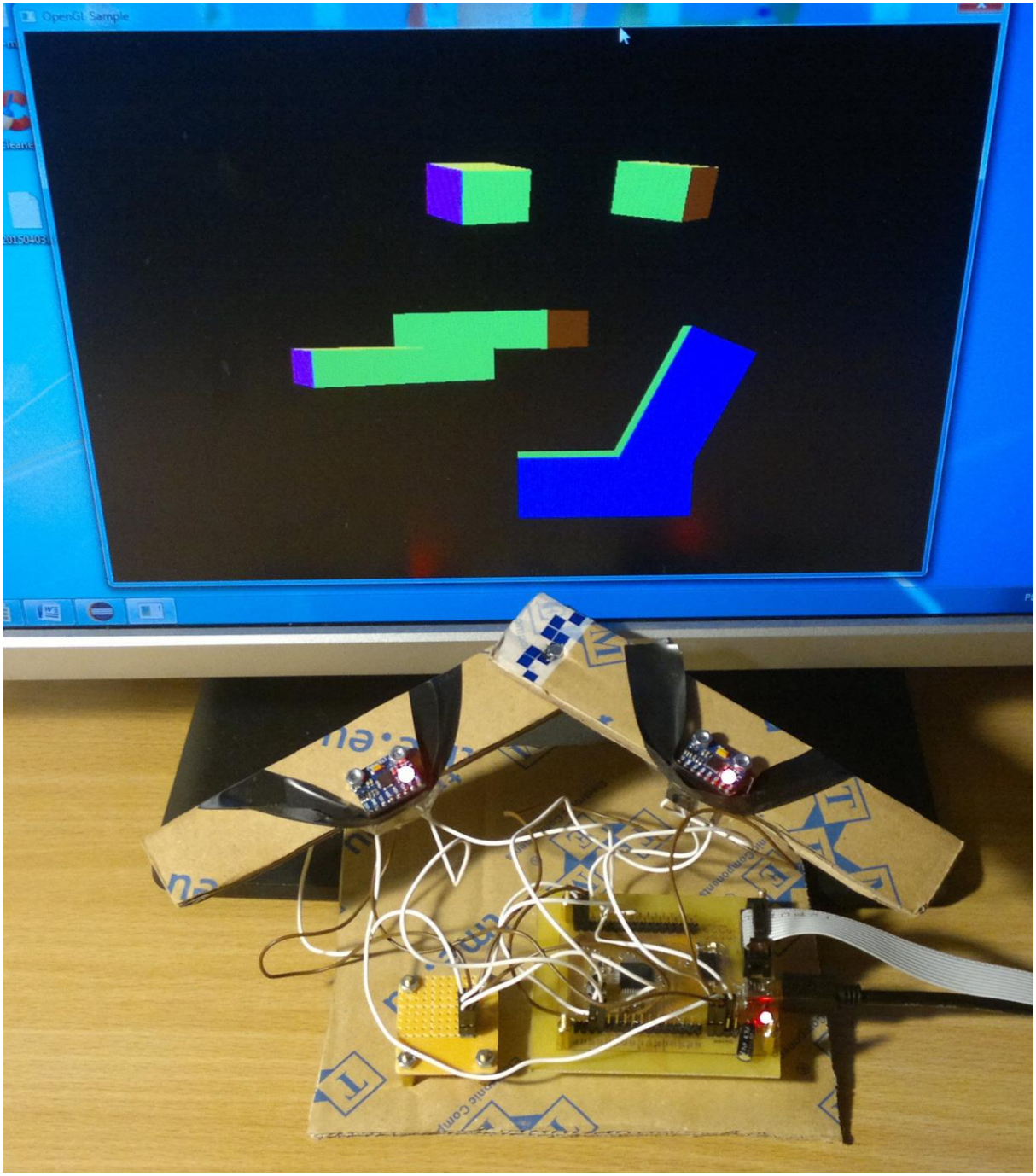
Funkcja zwraca zawsze mniejszy kąt pomiędzy żyroskopami.

Program załączę jako oddzielny wydruk.

Program wizualizujący dane został napisany w C++ z użyciem openGL. Wykorzystałem w nim jeden z pierwszych uproszczonych szkieletów z książki „Grafika. Fizyka. Metody numeryczne”.

Program załączę jako oddzielny wydruk.

5. Efekt końcowy



Rysunek 6 Działający układ