OBJECT ORIENTED PROGRAM IN VISUAL C++ TO DRIVE A SOLAR PANEL

POPA SORIN EUGEN[1*]

[1] *"Vasile Alecsandri" University of Bacau, Romania*

Abstract: The paper presents a program in Microsoft Visual C++ 2008, how control an USB interface board used to command a unipolar stepper motor. The stepper motor driver is made with a dedicated integrate circuit. This unit is used to control a photovoltaic panel solar orientation. The orientation angle is dictated by a sensor made of two photodiodes.

Keywords: Visual C++, stepper motor, solar orientation, photovoltaic panel

1. INTRODUCTION

The paper presents an object oriented program made in Visual Studio 2008 (Visual C++), which command a stepper motor used to drive a photovoltaic panel. The program consist in two modules, first, named "AutoCheck" how realize a self testing of the sensor and mechanical part, and, the second module, named "Normal" is the one how drive and orientate the photovoltaic panel to sunshine.

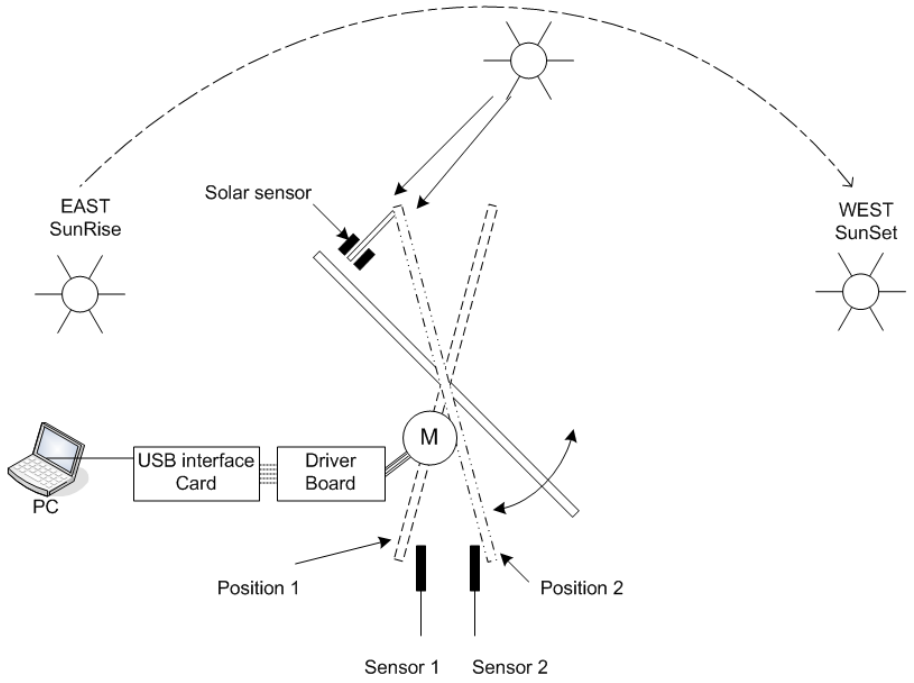

Fig. 1. The photovoltaic panel system orientation.

* Corresponding author, email: sorinpopa@ub.ro

In Figure 1 is presented the photovoltaic panel system orientation which will be led by the software application presented in this paper.

2. EXPERIMENTAL SETUP

2.1. Components of experimental devices

To accomplish the proposed task, was used an USB interface board, K8055. With this USB interface board is commanded the driver for stepper motor. The driver is realized around UCN5804B, translator/driver for unipolar stepper motors.

The K8055 USB interface board has the following input/output ports, used as shown in Table 1.

Table 1. The input / output ports of K8055 and their role in project.

Item No.	Port type	Port number	Used for	Specification
1	Digital input	1	Drive limit – sunset	Sensor1 = 1 – position 1 Sensor1 = 0 – going to position 2
2		2	Drive limit – sunshine	Sensor2 = 1 – position 2 Sensor2 = 0 – going to position 1
3		3 – 5	Unused	
4	Analogue inputs	1	Photodiode sensor 1	Data1 – east side
5		2	Photodiode sensor 2	Data2 – west side
6	Digital output	1	ON/OFF	1 – run / 0 – stop
7		2	Direction	1 – Sens1 = to sunset 0 – Sens2 = to sunrise
8		3	One phase	
9		4	Half step	
10		5 - 8	unused	
11	Analog output	1-2	unused	

More K8055 specification can be find at [1] and the driver for stepper motor is made according to UCN5804B integrated circuit, how can be find at [2].

3. RESULTS AND DISCUSSION

3.1. General presentation of application

In Figure 2 is presented the main Form of application, named “Solar v 01”.

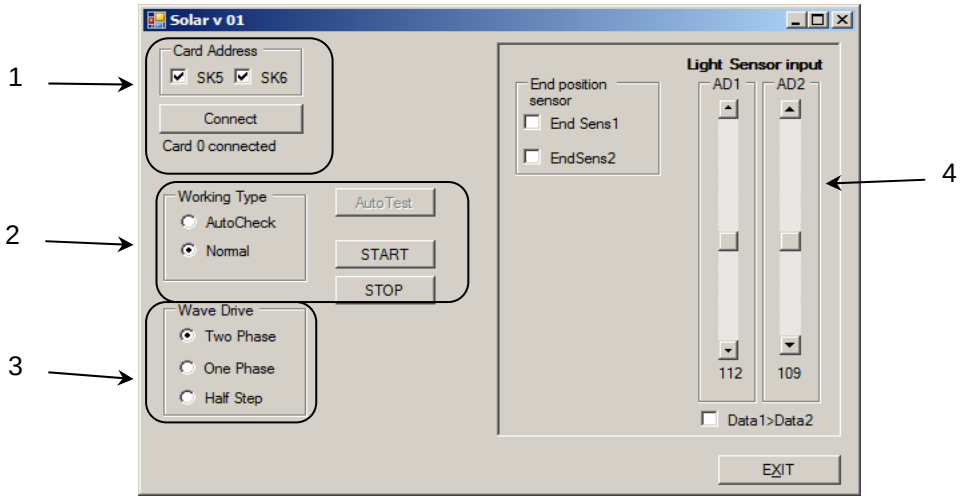

Fig. 2. Main front panel of application: 1-connection to USB module, 2-selecting the working type, 3-selecting the stepper mode for motor, 4-reading sensors values.

There are four distinct sections of applications: the first is responsible to detecting and connecting to USB interface board; could be connected until to four cards.

In the second section we can select between tow working mode, “Auto Check”, when is realized a self control of software and hardware mechanism, and “Normal”, where we can activate the application. The third section permits us to select the step for the stepper motor: two phase, one phase or half step. The fourth section is dedicated to showing us the sensors values. The drive limits sensors and the sunlight sensor.

3.2. Self test procedure

In this procedure, we test the end course sensors, and proceed to a complete angular rotation of photovoltaic panel, and execute a partial or total rotation from position 1 to position 2 and reverse. The final position will be position 1. To do that, we use two timers, 'timer3' and 'timer4'.

```
private: System::Void button3_Click_1(System::Object^  sender, System::EventArgs^  e)
         {     s1 = ReadDigitalChannel(1);
                       s2 = ReadDigitalChannel(2);
                        if ((s1==1) && (s2==1))
                                MessageBox::Show ("Sensor EROR ");
                        if (s1==1)
                               timer3->Enabled=true;
                        if (s2==1)
                               timer4->Enabled=true;
                        if((s1==0)&&(s2==0))
                               timer4->Enabled=true;          }
private: System::Void timer3_Tick(System::Object^  sender, System::EventArgs^  e)
        {//timer3 - duce in sens1 apoi lanseaza timer4 pentru sens2
        s2 = ReadDigitalChannel(2);
         if (s2==0)
               {
               SetDigitalChannel(2);
               SetDigitalChannel(3);
               }
         else
               {
                timer4->Enabled=true;
                timer3->Enabled=false;
                }     }
private: System::Void timer4_Tick(System::Object^  sender, System::EventArgs^  e) {
         //sens2 de mers
         s1 = ReadDigitalChannel(1);
         if (s1==0)
               {SetDigitalChannel(2);
               ClearDigitalChannel(3);}
         Else    { ClearDigitalChannel(2);
                   timer4->Enabled=false; }
         }
```

3.3. Detecting the solar orientation

Te determine the angular position of photovoltaic panel so the sunlight to fall normal to them, was created an hybrid sensor, made from tow photodiodes, separated by an solid panel, as shown in figure 3.

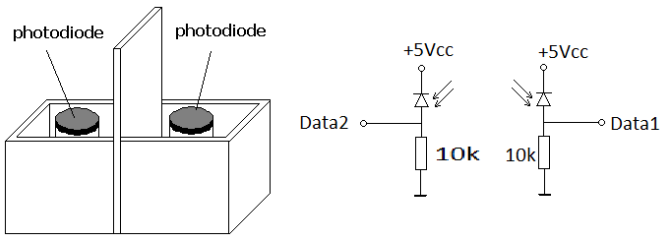

Fig. 3. The sunlight sensor.

These sensors are connected to analog input ports, and the values read are compared. If the voltage on port A1 is greater or less than port A2, then the panel are moved, otherwise the panel doesn’t move. When the value of voltage is too less, then the panel is moved to original point.

3.4. Normal work

When radio button "normal" is selected, are activated two buttons: START and STOP. The START button enable "timer2" how monitor the sunlight sensor and drive the photovoltaic panel according to. At the end of the day, when sun lights are poor, the voltage generated by photodiode is low, less than a predefined value, the "timer2" drive the panel to original position – sunshine position.

The "STOP" button disable "timer2" and stop device working. The timer2 code is presented:

```
private: System::Void timer2_Tick(System::Object^ sender, System::EventArgs^  e) {
       if (Data1>30)
       {      if (sens >5)
              {SetDigitalChannel (1);
              SetDigitalChannel (2);        }
              if (sens <-5)
              {SetDigitalChannel(1);
              ClearDigitalChannel(2);       }
              if(sens>=-5 && sens<=5)
              {ClearDigitalChannel(1);
              ClearDigitalChannel(2);       }
       }
       else {  s1=ReadDigitalChannel(1);
              if (s1==0)
              {SetDigitalChannel(1);
              SetDigitalChannel(2);  }
              else
              {ClearDigitalChannel(1);
              ClearDigitalChannel(2);
              timer2->Enabled=false; }
              }       }
```

4. CONCLUSIONS

The application permit a fully control of photovoltaic panel movement, a rapid intervention if something wrong happening and the possibility of implementing new subroutines allowing performance improvement. Adopted solution is suitable for future research in photovoltaic panel orientation, because the k8055.dll driver user by USB interface card permit until four cards simultaneous connected to one PC. So, we can connect supplementary sensors to efficiently monitor the electrical power production, the exact position of panel, and other inputs.

REFERENCES

[1] http://www.velleman.eu/distributor/products/view/?id=351346.
[2] http://pdf1.alldatasheet.com/datasheet-pdf/view/55138/ALLEGRO/UCN5804B.html.