Annals of the University of Petrosani, Electrical Engineering, 20 (2018)

DESIGN AND IMPLEMENTATION OF A 360° IMAGING SYSTEM AND ANDROID APPLICATION FOR REMOTE CONTROL

LORAND BOGDANFFY[1], GABRIEL ILCEA[2]

Abstract: The panoramic photograph has been used for some time, even before the digital era, the classic method being the merging of several photos to obtain a very wide field of view. With the advent of digital cameras, smartphones, much has been done to create photos with a wide field of view, even 360° panoramas, but sometimes, especially with phone cameras, quality is below acceptable.

This paper focuses on making a remote controlled rotating support system for a large DSLR camera in order to take high quality, very high resolution 360° panoramic images for businesses or institutions where a professional solution is needed.

Keywords: 360 imaging, panorama, android, software, hardware.

1. INTRODUCTION

This paper proposes an automated high quality panorama image creating system, composed from multiple images, controllable from any Bluetooth smartphone with Android operating system. The main advantages are:

- After the initial positioning and adjustment of the machine settings, a button is pressed and everything is done automatically, 150 photos being effortlessly made, the speed being approximately one picture at 2-3 seconds (30 pictures per minute, 150 pictures in 5 minutes) at least 10 times faster than manually repositioning the camera. The advantage of short a time span between consecutive frames significantly reduces the possibility of scene changes that may occur during capture of images.
- Control of accurate device rotation and automatic capture minimize the risk of errors, such as missing certain areas in the panorama.

The hardware part of the application was made by using largely recycled components such as stepper motors and toothed wheels from disused printers and other parts that I adapted to my installation.

[1] *PhD., Assistant Prof., Eng., University of Petroşani*
[2] *Ph.D., Eng., Energy Complex Hunedoara*

2. CONCEPT AND IMPLEMENTATION

Panoramic photography is a technique of photography, using specialized equipment or software, that captures images with horizontally elongated fields of view. It is sometimes known as *wide format photography* [2].
Required components for obtaining a stitched panorama (from multiple images):

1. Camera;
2. Tripod, tripod head;
3. Stitching software, available online.

Additional components for automating the process:
- Hardware components:
  - PIC16F88 or PIC16F877 Microcontroller;
  - BTM-222 Bluetooth Module (Class 1) for remote communication;
  - Mitsumi M42SP-7 stepper motor for driving the assembly;
  - CNY-74 – optocoupler for isolation of the camera's tripping circuit from the electrical side of the mounting;
  - ULN2003A Integrated with 7 Darlington gates for step-up motor amplification, each with a capacity of 500mA, for driving the motor.
- Software:
  - Microcontroller program;
  - Android program, for controlling the system.

The camera, that can be triggered electronically, shown in the next part.

In order to trigger the DSLR camera shutter, you need to close two of the camera's circuits, one for auto-focus, the other for exposure. The manual triggers function as a switch that activates each stage. (this was tested o a Nikon camera)

Table 1. ***Trigger the DSLR***

Trigger logic		
Focus	Trigger	Result
0	0	-
0	1	-
1	0	Camera focuses
1	1	Camera shutter is released

The use of an optocoupler, and two gates are needed to perform the focus / trigger operation.

Features of CNY-74

- CNY74-2 includes 2 insulating channels;
- CNY74-4 includes 4 insulating channels;
- DC test voltage of insulation VIO = 2.5 kV;

- Test class 25/100/21 DIN 40 045;
- Low coupling capability of 0.3 pF typical;
- Current Transfer Ratio (CTR) typical 100%;
- Low CTR coefficient;
- Wide range of usable ambient temperature.

The PIC16F877 microcontroller controls tripping and is also used for stepper motor control.

Communication with the Bluetooth module is achieved by linking the RX / TX pins of the microcontroller to the TX / RX pins of the Bluetooth module.

Optionally to reduce the size is enough to use an 18-pin microcontroller like PIC16F88 that has a RS232 hardware port.

The Mitsumi M42SP-7 is a single-pole stepper motor with a common yarn and four windings, which takes 48 steps to complete the rotor's rotation.

Features of the M42SP-7 motor

- Rated voltage - DC 12V;
- Usable voltage - DC 10.8 ~ 13.2V;
- Rated current / phase - 259mA;
- Number of phases - 4;
- DC coil resistance - 50Ω / phase ± 7%;
- Step angle - 7.5˚;
- Driving Method 2-2 (Unipolar);
- Class isolation - Class E;
- Holding torque - 49.0mN · m;
- Pull-out couple - 23.5mN · m / 200pps;
- Pull-in torque - 19.6mN · m / 200pps;
- Frequency max. pulse pulse-out - 600pps;
- Frequency max. pulse pulses – 420pps.

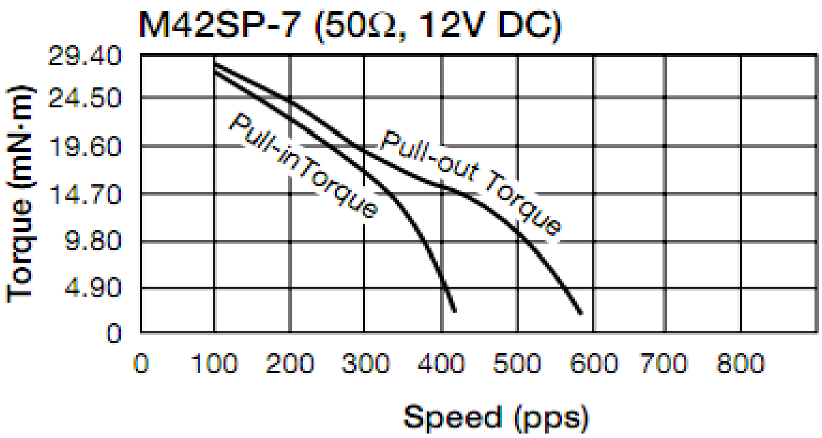

Fig.1. Torque/Speed graph

Between the motor and the microcontroller we used the integrated ULN2003A that has the following features:

- Max. collector-emitter 50 V;
- Input voltage, VI 30 V;
- Collector maximum current 500 mA;
- Output current limitation, IOK 500 mA;
- Emitter-terminal total current -2.5 A.

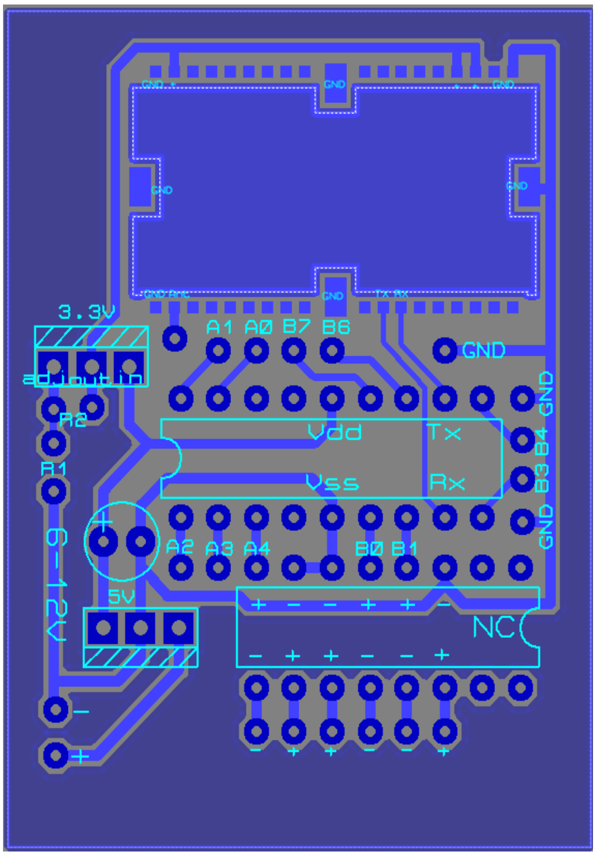

Fig. 2. Complete hardware trace design

A number of four commands were programmed to be transmitted:

• Shutter - with parameters: focus time (0-99 seconds / 10), exposure duration (0-99 seconds), number of exposures (0-999), exposure delay (0-99 seconds);

• Rotation - with parameters: Number of steps (1-9999, one step means 0.2 degrees, one degree = 5 steps, a complete rotation = 1800 steps), delay between steps (0-99 milliseconds), and last but not least (trigonometric and vice versa);

• Auto-panorama - with parameters: total angle (degrees), focal length of the lens (mm), delay time before exposure (0-99 sec / 10) and of course the direction;

• Cancel - at any time if the Cancel button is pressed, the execution of any command is interrupted.

Due to this communication method, once a command is sent, it is executed by the MC even if the wireless connection is interrupted for various reasons.

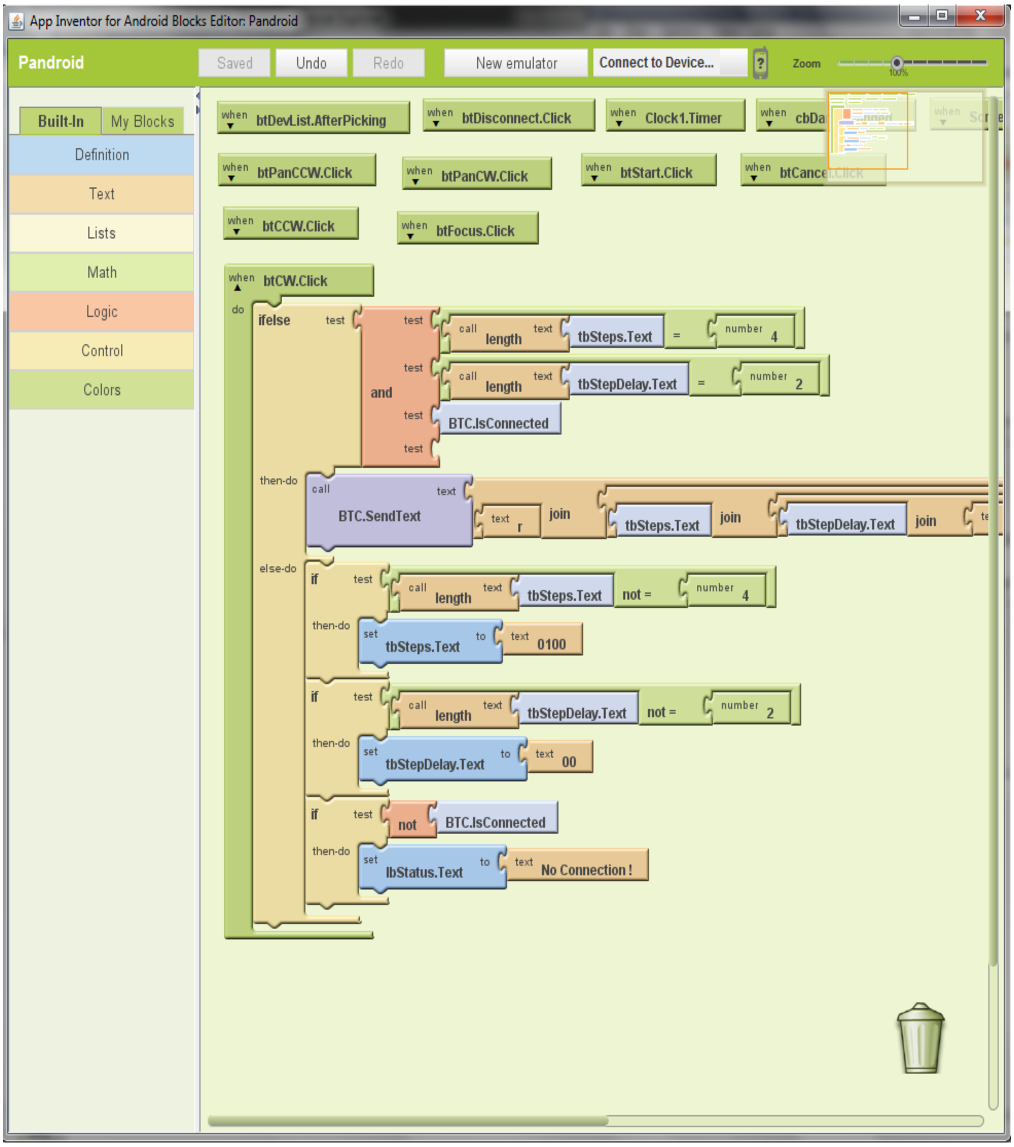

Fig.3. Appinventor complete interface, and click function

Before any transmission can occur between Android Smartphone and microcontroller, the devices must be paired, and a connection must be established. This is the first priority, after powering up the devices.

After that, the controller listens to input from the smartphone, and recognizes the four commands mentioned before. Some of which come with parameters. An example of such a command is given in figure 3.

A good practice is using the microcontrollers interrupts so that receiving commands will work even if the µC is busy carrying out its main task. This is especially important to be able to cancel a running task.

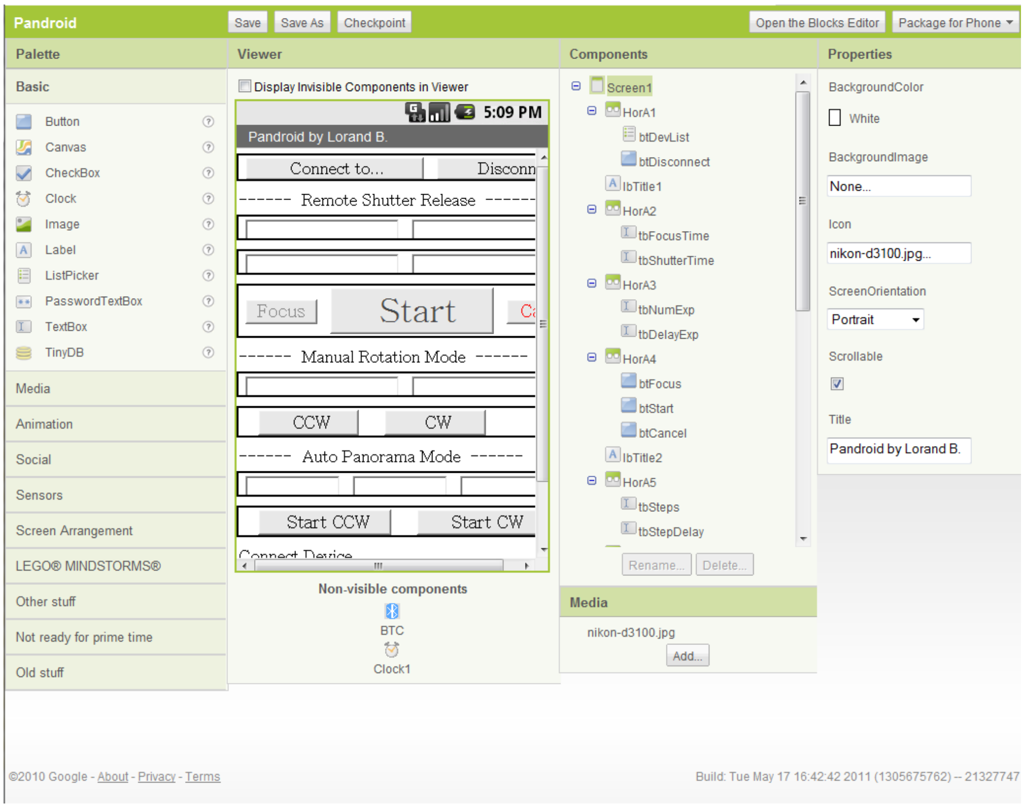

Fig. 2. Designing the interface

```
#int_RDA
void  RDA_isr(void)
{
  ch=getc();
  if (ch=='c')
  {  cancel=1;
     return;
  }
  if (executing==0) switch (ch)
  {
          case 's':{start=1;
                break;}
          case 'm':{startm=1;
                break;}
          case 'a':{starta=1;
                break;}
          default:{data[nrChars]=ch;
                nrChars+=1;
                break;}
  }
```

The above section of code reads and recognizes one of the 4 types of commands. Because it has to run very fast, real-time, as it is attached to an interrupt, the instructions for each command are described in separate functions that are called from within the function handling the interrupt.

The next section is a function to manually position the system. After a parsing sequence, according to the data received, the motor is driven in the direction and to the angle required.

```
void manualMove()       //-------------------
manualMove------------------------
{
  if (nrChars!=7)
  {  data="";
    nrChars=0;
    return; }

  int16 i=0;
  executing=1;

//    Start data parse
  direction=data[0];
  steps=steps+((int16)(data[1]-48))*1000;
  steps=steps+((int16)(data[2]-48))*100;
  steps=steps+(data[3]-48)*10;
  steps=steps+(data[4]-48);
  wait=wait+(data[5]-48)*10;
  wait=wait+(data[6]-48);
//    end of data parse

  nrChars=0;        // resetare nrChars
  data="";          // resetare data

  if (direction=='l')
  {
    for (i=1;i<=steps;++i)
      {
        if (cStep==step4) cStep=step1;
        else cStep=cStep+1;
        output_high(cStep);
        delay_ms(10);
        output_low(cStep);
        if (wait>0) delay_ms(wait);
        if (cancel==1) break;
      }
  }
  if (direction=='r')
  {
    for (i=1;i<=steps;++i)
      {
        if (cStep==step1) cStep=step4;
        else cStep=cStep-1;
        output_high(cStep);
        delay_ms(10);
        output_low(cStep);
        if (wait>0) delay_ms(wait);
        if (cancel==1) break;
      }
  }

  steps=0;
  wait=0;
  startm=0;
  executing=0;
}
}
```

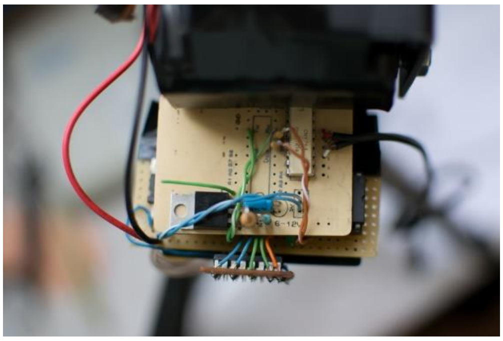

Fig. 5. System close-up, top view

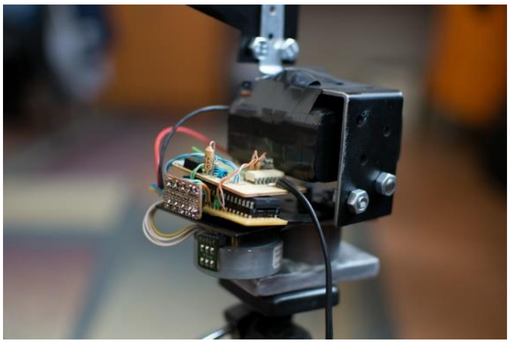

Fig.6. System close-up, side-view

3. CONCLUSIONS

The prototype system lacks a professional mechanical build, but that was not the main purpose, instead we managed to create a communicating system from mostly scrap and recycled material.

Other than that, it’s a perfectly usable system, in a real environment as shown with the below results. The programming is the strong point of the system, and required the most effort. The complete program isn’t presented in the paper.

To create the final image, from the many images taken with the system, a software is needed for stitching. We chose “Panorama Tools“, an open source solution for stitching large panorama images.

The images obtained, if created on a high performance PC, with at least 16 GB of RAM, can be as large as in the scale of GigaPixels.

Fig. 3. Spherical panorama obtained with the system

REFERENCES

[1]. **Jones D. W.**, *Control of Stepping Motors*, 1995 **http://www.yertiz.com/cnc/steppermotor.pdf**

[2]. **Kang S. B., Benosman R.**, *Panoramic vision: sensors, theory, and applications*, Springer, ISBN 978-0-387-95111-9, 2001.

[3]. **Pal S., Tripathy N. S.**, *Remote Position Control System of Stepper Motor Using DTMF Technology*, International Journal of Control and Automation, Vol. 4, No. 2, pp.35-42, June, 2011.

[4]. *** *Documentația tehnică a circuitului integrat ULN2003A*, Texas Instruments, 2016.