PC User Interface of Arduino Dome Control Board

# Getting Started

## Introduction

This application is a graphical user interface for a microcontroller board based on Arduino technology. This application is meant to control the dome rotation in function of a telescope position.

This is a very light application consuming little resources and taking very little memory.

## System Requirements

System requirements for the Arduino\_Dome application are pretty little:

* Pentium III processor;
* 1GByte RAM;
* Microsoft .NET 3.5 Framework;
* 10MB Free space on hard-disk;

## License

# Software Architecture

## Firmware Architecture

The firmware running into the Arduino board is basically a command shell. It inspects the USB Channel (configured as Communication Device Channel CDC) awaiting for a string containing the command to execute and its parameter if needed. It executes then the command and returns the result.

To note that most of the commands perform an action on the board therefore they might not returns any information back to the PC, nevertheless to create a more robust application it is introduced a sort of ACK just to tell that the command was terminated without any error. In case of error the error message is sent back to the application.

The firmware might be seen as the Server into a Client-Server system. Here is the list of available commands (correct syntax will be further described) :

* Turn Clockwise : slew clockwise the Dome ;
* Turn Anticlockwise: slew anticlockwise the dome;
* Stop: stop slewing the dome;
* Position: get dome position (as circular counter);
* Set Position: set the dome position (as circular buffer);
* ACK: ask acknowledge to the device (check is connection is still up and running);
* Gear Configuration: set the dome gear multiplication factors;
* Get information: about the RTOS and the firmware;
* Help: help message.

## Dome Position Codification

Dome position is read thanks to an incremental optical encoder. However, since there is no 1:1 relationship between the encoder full rotation and the dome full rotation it is necessary to carry out the actual position based on the previous encoder position and the multiplication factors. Thus to indicate the dome position it is chosen a simple system:

* It is first carry out the number of steps to make a full 360° dome rotation: this is simply the encoder steps count times the encoder / dome gear ratio,
* Then a counter keeps track of the position: each “hit” of the encoder it is incremented or decremented in function of the rotation direction;
* If counter reaches the counting value to make a 360° rotation it is set to 0, and if it reaches 0 it is set to the max counting value;
* Angle position is given by this proportion: angle / 360 = counter value / MAX counter value.

## Application Architecture

The application is the client of the Client-Server paradigm. It generates some requests that are sent to the microcontroller board. The microcontroller board executes the request and eventually provides back some results that will be consumed.

The communication link between the PC and the microcontroller board is made via USB, whose port is configured to be managed as a Serial Port. Therefore all the COM-like commands are used for the communication between the microcontroller board and the PC.

PC application is based on the ASCOM standards libraries to ease the connection with all the photography instruments, telescope in particular. To note that for the time being this application does not manage the dome shutter.

PC application is organized as follow:

* A driver for the Arduino microcontroller;
* An ASCOM Dome driver to ease export;
* ~~A light ASCOM telescope library to interface with the ASCOM telescope present in the system;~~
* The graphical UI for the human interaction.

## Application Use

The application may be used in two ways:

* As manual dome rotation control;
* Synchronize with the telescope position.

In the latter case the application periodically senses telescope position and adapt the dome position in a way that the difference between the two is below a fixed threshold. Until this condition is fulfilled no rotation is need. If condition is not fulfilled a new rotation will be done.

To note that for safety reasons, the application will not turn the dome if it senses that the telescope is moving. This limitation will ensure that the dome is not running behind the telescope and preserve the AC motor used for dome rotation. Therefore it waits that the telescope has finished its slewing before starting to move. Drawback is a delay in having the system ready to shoot estimated to be one minute worst case.