**Timing Controller Software Specification**

*April 5, 2019 - Arthur Bondar*

Controller

For this application, a small and easy to use Arduino Pro Mini (ATmega 328P based) have been chosen. Arduino is a widely supported IDE with a large ecosystem for with varieties of libraries for various sensors.

Application

The main Raspberry Pi Zero SBC is quite powerful but limited in its capacity to interface external sensors and hardware. In addition, the main SBC will be powered down the majority time of deployment to reduce power consumption. A small low power MCU board will schedule wake and sleep cycles and interface with the necessary sensors.

Connections

The MCU will be connected to a 5v output of the boost converter and use a MOSFET to switch the power for the main computer. All sensors connected to this system must either be 5v tolerant or use an open drain bus output. For communication, the control will use UART (Serial) to communicate with the Raspberry Pi and I2C/OneWire protocol to connect to various sensors.

API

The interface is implemented as a simple serial console with GET, SET and HELP commands. As the name suggests, GET will return the variable and SET will write new data to a sensor/device. HELP will be used to dump all the sensor data form debugging. By using the serial console, the code can be easily developed and debugged on the board separately. When the software is ready, the Pi will interface with the Arduino just the same as the developer using the Arduino Serial console.

Sensors

The Arduino is connected to the following hardware:

* MOSFET for power switching (GPIO)
* LED for light indication (GPIO)
* Battery voltage (Analog GPIO)
* RTC for timekeeping (I2C)
* Temp sensor (I2C)
* INA219 for power monitoring (I2C)
* Memory buffer (internal mem storage

Sequence

When the user (or Raspberry PI) connects to the Arduino via Serial 1, the user can send a backslash ‘/n’ character to get a screen with commands:  
**please use GET, SET followed by a command or HELP to get a list of parameters**

List of commands HELP print:

|  |  |
| --- | --- |
| **GET firmware** | Version and time/date of the firmware |
| **GET fullstart** | Time since power up and the start of deployment |
| **GET start** | Time of day when Pi is turned on |
| **SET start <hh:mm>** | Set the time when Pi needs to start |
| **GET end** | Time of day when Pi is turned off |
| **SET end <hh:mm>** | Set the time when Pi needs to end recording |
| **GET remaining** | Time remaining in the day for recording |
| **GET mosfet** | Status of the power switching mosfet |
| **SET mosfet <0 or 1>** | Open or close the power switch |
| **GET time** | Time from the onboard RTC |
| **SET time <YYYY/MM/DD hh:mm:ss>** | Set time on the RTC |
| **GET led** | Status of an LED |
| **SET led <0 or 1>** | Turn the LED on/off |
| **GET temp** | I2C/OneWire temp sensor data |
| **GET current** | Current from INA219 |
| **GET voltage** | Voltage from INA219 |
| **GET power** | Power from INA219 |
| **GET batt** | The voltage of the battery |
| **GET mem <num>** | Get the Internal memory storage when Pi is off |
| **SET mem <num>** | Store a value in the internal buffer |
| **SET clearmem** | Clear all data in memory |