

1 Introduction

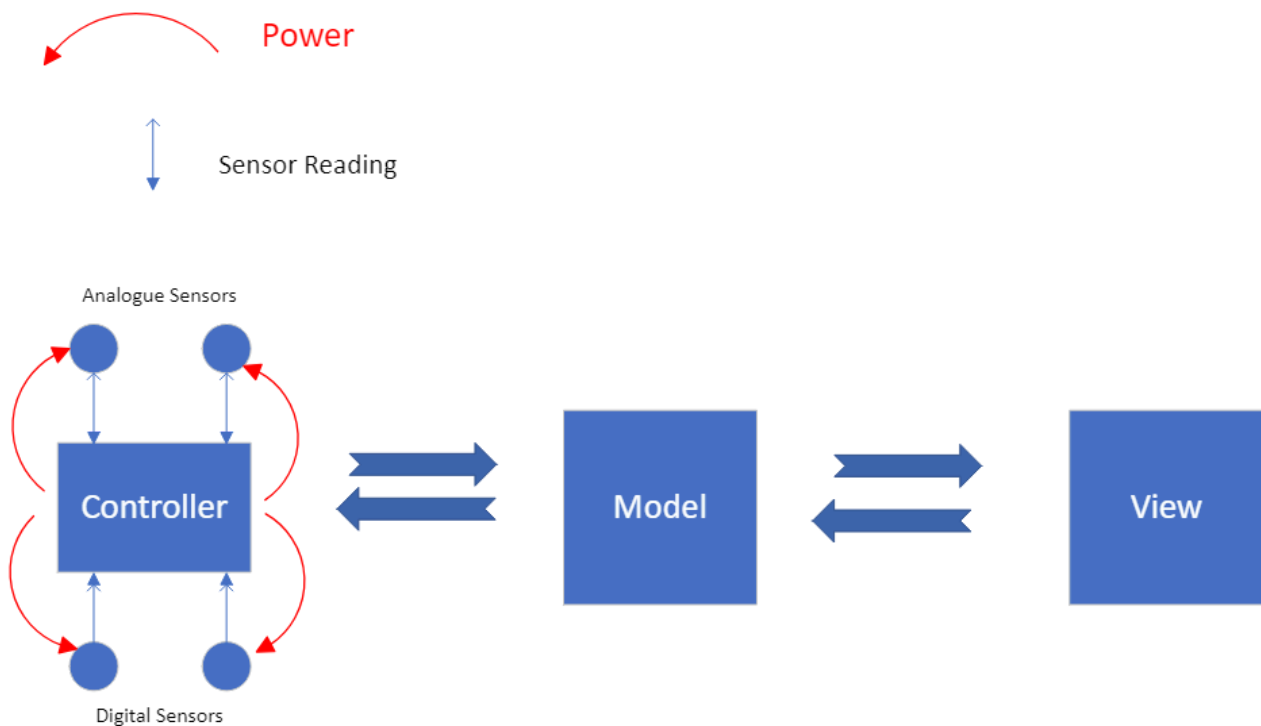
Write software to communicate remotely with a fictional hardware controller board using Python. Model the state of the hardware, and provide an interface to view data in the model. The form of the interface can be text based or graphical.

An actual server representing the controller does not have to be implemented, and the communication can be mocked. The communication protocol and commands are provided below, and methods for sending commands and decoding the responses should be demonstrated.

Demonstrate how the model and view change based on commands sent to the controller, and the response data that is received.

Some comments on the hardware are provided as optional things to consider about how the hardware might function as inputs and outputs are changed, and how data in the model would change based on certain commands and responses.

As much or as little of the hardware behaviour and protocol can be implemented as desired.



2 Hardware Overview

The controller board has 4 sensors connected; 2 digital and 2 analogue. Signals from these are connected as inputs to the controller board.

The controller has 4 digital outputs which can switch on power supplies to each sensor. In addition, there are 4 analogue outputs which indicate the power being supplied to the sensors.

2.1 Hardware Specification

Function	Number	Channel	Comment
Digital Outputs			
Sensor power supply	4	0-3	0 switches a sensor power supply off 1 switches a sensor power supply on Channel 0-1 map to digital sensors 0-1 Channel 2-3 map to analogue sensors 0-1
Digital Inputs			
Digital sensors	2	0-1	Digital sensor 0 indicates if light is detected Digital sensor 1 indicates if motion is detected
Analogue Outputs			
Supplied power	4	0-3	These output the power being consumed by the respective sensor. Channel 0-1 map to digital sensors 0-1 Channel 2-3 map to analogue sensors 0-1 An output power of 20 W is provided to a sensor that is powered on. None is provided otherwise.
Analogue Inputs			
Analogue sensors	2	0-1	Analogue inputs from sensors to the controller: Channel 0 – Power detected by the light sensor in Watts Channel 1 – Distance from sensor that motion is detected in millimeters

A basic communication protocol is provided below. It represents the communications with the hardware controller board.

3 Communications Protocol

The general form of a command message is an ASCII human-readable string starting with a caret (^) and ending with a newline (\n) character. Command strings follow a strict format with each character and space being significant. A typical string consists of a caret, followed by a command letter, a sequence number, possibly a number of parameters, and the trailing newline. For example, the command to set an output has the form

```
^O 2A AO04 8C21\n
```

where “O” is the command letter for the output command, “2A” is a hexadecimal sequence number, “AO” indicates an analogue output is to be set, “04” is the hexadecimal output number, and “8C21” is the 16-bit hexadecimal value to set on the output.

The general form of the reply message to a command is similar to the above, having the same start and end characters and the same general form. The body of the reply consists of the command letter and sequence number repeated back unmodified from the original command, followed by a three-letter result code, then a number of parameters depending on the command itself. As an example, the response to the command above, assuming that there is actually an analogue output four would be

```
^O 2A OK_ \n
```

where the three letter result code in this case is “OK_” indicating that the command was successful. In this case the reply passes back no parameters.

4 Controller Commands

The following table summarises the commands by command letter.

	Command	Description
E	Echo	Echo request.
I	Input	Read and return the state of an input or output.
O	Output	Set an output state.
P	Power Up State	Set the power-up output state.

In the following descriptions, the text formatting as the following significance:

- The start (“^”) and stop (“\n”) characters are not shown in the descriptions below, but they should be prepended and appended to the strings without any intervening white space.
- Characters in fixed upright indicate fixed and case sensitive parts of a string.
- Parameters in *italics*, such as *tttt*, indicate a variable parameter and the fixed number of characters in the parameter.
- In all commands the sequence characters are indicated with SS, and they are arbitrary and returned as-is to the master in the message reply.

There are return codes that can be sent to the master in a reply from the controller. These consist of the three-character string and can have any of the following values:

OK_ Command and parameters were acceptable, and the command has been executed.

ERR There was something wrong with the format of the command.

RNG Message was valid and the command was valid and correctly formatted, but a parameter was out of range, for example trying to set a non-existent output channel.

4.1 Echo

Echo a reply to the master.

Command string

```
E SS
```

Return string from controller

```
E SS rrr sss
```

Parameters

SS Sequence number

rrr Return code

sss System state

Description

The echo command is used by the master to verify that the controller is on-line.

The system state returned by the echo command can be one of the following:

PWR The system is in the power-up state.

OFF The system is in the off state.

1 **Example**

```
^E 00\n
```

```
...
```

```
^E 00 OK_ PWR\n
```

4.2 Input

Read the value of a digital or analogue input, or return the current output value on a digital or analogue output.

Command string

I SS <i>x dnn</i>

Reply string from controller

I SS <i>rrr Adnn vvvvvvvv</i>	//Analogue
I SS <i>rrr Ddnn w</i>	//Digital

Parameters

SS Sequence number

x Input or output type – “D” for digital, or “A” for analogue

d Input or output direction – “I” to select an input, “O” to select an output

nn Two digit hexadecimal number indicating the IO channel to be returned, “00” to “FF”

vvvvvvvv Eight digit hexadecimal representation of the floating point value of an input or output

w 0 or 1 indicating current value of the digital input or output

rrr Return code

Description

The input command returns the latest input or output value from the IO point. The values returned from input channels reflect the latest read values from the hardware.

1 Example

^I 01 AI05\n
...
^I 01 OK_ AI05 000F999B\n

4.3 Output

Set the value of a digital or analogue output.

Command string

```
O SS AOnn vvvvvvvv //Analogue
```

```
O SS DOnn w //Digital
```

Reply string from controller

```
I SS rrr
```

Parameters

SS Sequence number

xx Output type – “DO” for digital, or “AO” for analogue

nn Two digit hexadecimal number indicating the IO channel to be returned, “00” to “FF”

vvvvvvvv Eight digit hexadecimal representation of the floating point value of an output

w Digital output value to set, 0 or 1

rrr Return code

1 Example

```
^D 03 DO01 1\n
```

...

```
^O 03 OK_\n
```

4.4 Power Up State

Set if the controller board should be fully powered up.

Command string

P SS *w*

Reply string from controller

P SS *rrr*

Parameters

- SS** Sequence number
- w* Set power up state, “0” for OFF, “1” for power up (PWR)
- rrr* Return code

Description

Set if the controller board should be fully powered up. All power supplies are disabled unless in the power up state.

Example

^P 04 1\n

...

^P 04 OK_\n