## **Use Python on a Raspberry Pi to display a GUI reading and writing to a device over modbus.**

Tendo is a startup company developing new types of flow sensors. We need short-term software development work. Your experience should include developing Python scripts, simple GUIs and communicating with serial devices.

Technical Description:

I am looking for a Python script that displays a touchscreen compatible GUI. The GUI communicates over modbus with a data capture device. I have existing example code in Python using PyQt that works under Ubuntu but it is only usable on a desktop monitor.

The example code I have currently displays a graph of 8 plots as well as a data display with numerical values and boxes to enter commands such as start, change calibration settings, etc. The current code uses minimalmodbus to communicate serially with our data capture device. The data capture device does all the data processing, the GUI needs to read values from the data capture device registers and sends some user inputted values back to the device (a command to start and calibration settings).

I need a less cluttered interface with a few commands that can be controlled with buttons (mouse clicks). The interface should run in full screen if possible.

The new code can either be modified from my existing code or created from scratch whichever is easier.

The hardware the code will need to run on is a Raspberry Pi 4B 4GB RAM and the Raspberry Pi 7” Touch Screen display - 800x480.

Existing example code and a register database with addresses and details will be provided. I’ll need to verify the code works with the data capture device before project completion without providing the actual device.

A screenshot of the existing example code running and an example of what I’d like done has been attached.

Final Deliverable:

Python code that communicates with our data capture device and displays a touchscreen compatible GUI.

Timeline: 2 weeks (10 business days M-F) after project start max.

Edit Apr 3rd 2023:

I've attached a more detailed description of the 2 screens I need in the GUI with PowerPoint slides and a word document describing the necessary features and how they interact with the serial registers. The GUI can look different than the described slides, using any built in buttons, fonts, themes, and appearances is preferred.

I've also attached the existing GUI code (tendo.py) the .whl file it installs in, and the register database (Database.csv)

Dear [Client's Name],

I hope this message finds you well. I am writing to provide an update on the GUI development project using PyQt and the Modbus RTU protocol for your capture device. I have completed building the GUI based on your description. However, there are a few points that I would like to clarify to ensure that it meets your expectations. If there are any changes that you require, please let me know, and I will be happy to make the necessary modifications.

Regarding the first page:

I have implemented the functionality to change the "amount of pour" using pre-defined values for each register. Each time the user presses the button, the value cycles through the list of values: 0, 0.75, 1, 1.5, 2, or 3.

Additionally, When the GUI first loads, it does not change the default value of the POUR\_CONTROL register. However, when you press the start button, the POUR\_CONTROL register is set to 1, as per your description, but you can't change the value to 0 ,2 or 3 unless you go to settings page,

Moving on to the settings page:

I have optimized the code by dispensing the CSV file and pre-defining all the necessary register names, addresses, and other related information directly in the code. This should result in faster performance since the program no longer needs to read the CSV file.

Also, as per your description, the tables display the initial values of the registers when the screen first loads. The program reads the values of all the needed registers from the capture device and displays them. If you meant that the program should set the default values by itself when it first loads on the capture device and then display them, please let me know.

In terms of design, I have opted for a simple dark theme to make it easy on the eyes. The buttons are also large, ensuring ease of use with the touch screen.

Please feel free to review the current implementation and share your feedback. If you would like anything changed, don't hesitate to let me know, and I will be happy to make the necessary modifications.

Looking forward to hearing from you.

Best regards,

[Your Name]