Progress Report

CCTVal

CENTRO CIENTÍFICO
TECNOLÓGICO
DE VALPARAÍSO

December 9, 2022

Revision 4.1

1 Objectives and Specifications

We are assuming that a transmitter and a receiver need to be controlled through a WLAN that interconnects a central computer with Tx and Rx terminals, both of which are managed by a microcontroller (see Fig. 1). Our work focuses exclusively on the Rx terminal, its control and the reliable storage of measurement data. We do not include control of the frequency synthesizers. The objectives are as follows:

- Control of two stepper motors to select azimuth and elevation movement of channel-sounder Rx antenna. The specified hardware is Arduino Portenta with its corresponding breakout board.
- Maximum azimuth resolution: 6400 step/turn, with user selectable reduction by factors of 2^n.
- Azimuth speed $>$ 100 RPM with user-selectable soft-start and soft-stop speed (acceleration and deceleration).
- A/D conversion of input voltage once per angular resolution step. Time to store measured data must occupy no more than 5% of azimuth rotation time.
- Real time display of acquired angular power spectra on laptop connected to Portenta board.
- Graphical User Interface (GUI) running on laptop can control azimuth range, azimuth speed, and elevation range.
- Initial angular reference position of stepper motor set via GUI.
- Saved data files must contain all relevant settings used in the acquisition routine.
- Access of stored data files through LAN/WLAN.

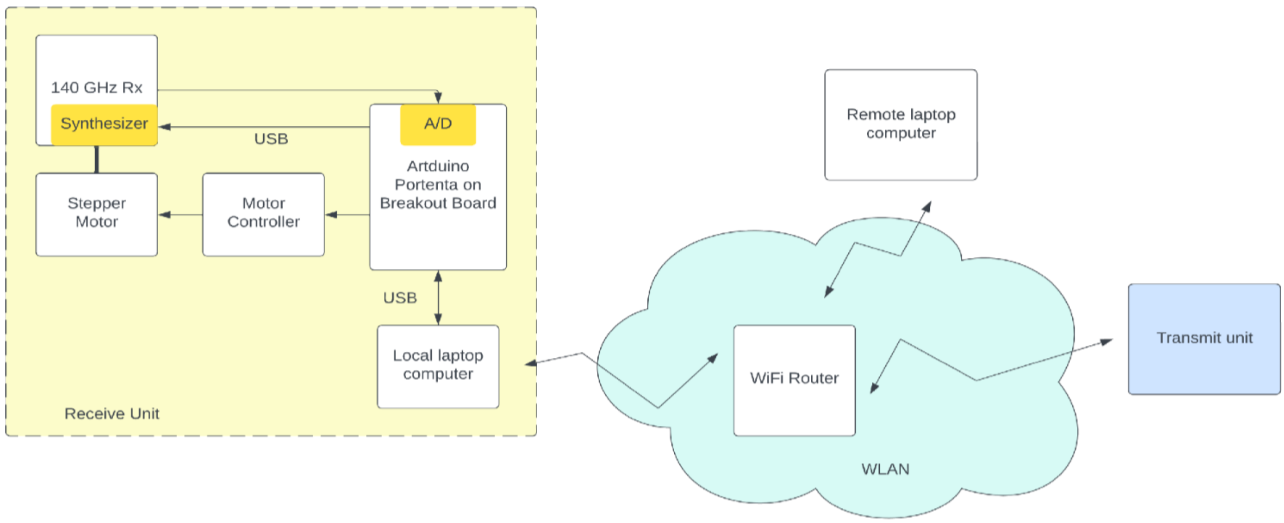

Fig. 1: Block diagram.

2 Results

- The program running on the Arduino Portenta controls azimuth and elevation motors in user selectable angular resolutions as specified in the objectives. The driver resolution is always set at its maximum of 6400 step/turn, and lower resolutions are emulated by skipping steps for storing data.
- GUI running on local laptop connected via USB to Portenta allows choice of operating parameters including initial reference position. This can be set manually by commanding movements independent of the data acquisition routine.
- Storage and reading of data from SD card on Portenta board proved difficult and was abandoned in favour of storage on locally connected laptop, which is fast enough for this purpose (480Mbps USB link). Stored data to be shared with central computer over WLAN.

- Azimuth speed is greater than 100 RPM for all angular resolutions, even added delays for acceleration and deceleration (see Table 2).
- A/D conversion of input voltage at a rate of once per angular resolution step. Time to store data of a complete azimuth scan occupies no more than 1% (about 0.7%) of azimuth rotation time using USB-connected laptop.
- Real time display of acquired angular power spectra on USB-connected laptop once per azimuth rotation.
- Remote wireless access of data stored in laptop via LAN/WLAN is feasible.
- Saved data files contain all relevant settings used in the acquisition routine (see Table 1).

3 Graphical User Interface User Guide

The developed user interface is shown in Fig. 2, where it has been divided into different sections to be explained independently. Each of the following subsections describe the corresponding number in Fig. 2.

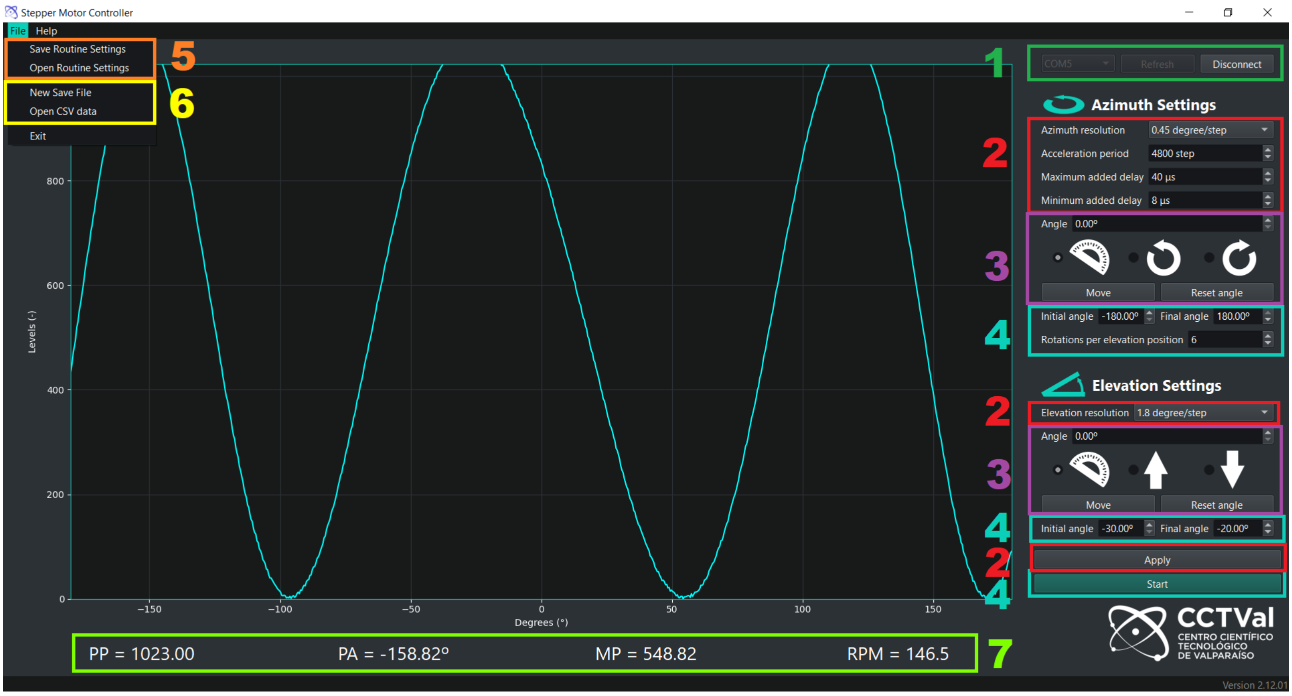

Fig. 2: Graphical User Interface mock-up

3.1 Connection

The connection section provides a list of available serial ports, which can be updated by clicking the **Refresh** button. Clicking the **Connect** button triggers the dispatch of a message with default parameters to the Portenta board, which responds with an acknowledgement message if the connection is established successfully. The absolute angle of azimuth and elevation and also set to zero. By default, the message sent has the following structure: "c−6400−4800−40−2−200−4800−40−8", where the first four numerical values correspond to azimuth resolution (in step/turn), acceleration period (in steps), maximum and minimum added delay (in microseconds). The next four values follow the same pattern, but corresponding to the elevation stepper motor.

Once the connection is established by receiving the acknowledgement message correctly, the **Connect** button changes its name to **Disconnect**, and all widgets are enabled. The **Disconnect** button sets the variable related to the port to `None`, and widgets are disabled.

3.2 Movement Parameters

These widgets receive different parameter values from the user to be used in the movement of the motor. The first required parameter for both the azimuth and elevation settings is the movement/measurement resolution, measured in degrees/step. This parameter has a number of fixed values given by the driver resolution options, ranging from

0.05625 to 1.8 degrees/step. As an example, if the 1.8 degrees/step resolution is chosen, a measurement will be made every 1.8 degrees. In order not to change the driver settings manually, these resolutions are emulated in software by always having the maximum resolution set at the driver, and skipping measurements accordingly in powers of two.

The rest of the parameters are related to the acceleration/deceleration routine. This is a necessary step to avoid motor malfunctioning in certain resolutions. The idea is to gradually change the motor velocity at the beginning and end of its movement, as seen in Fig. 3, where the time intervals (which can be thought as time delays induced to change the velocity of the motor) between movements are shown. To implement this, three different parameters are used:

- Acceleration Period (P_a): amount of steps used to gradually change the velocity at the beginning and end of the movement.
- Maximum Added Delay (T_{as}): maximum induced delay between movements.
- Minimum Added Delay (T_{ai}): minimum induced delay between movements.

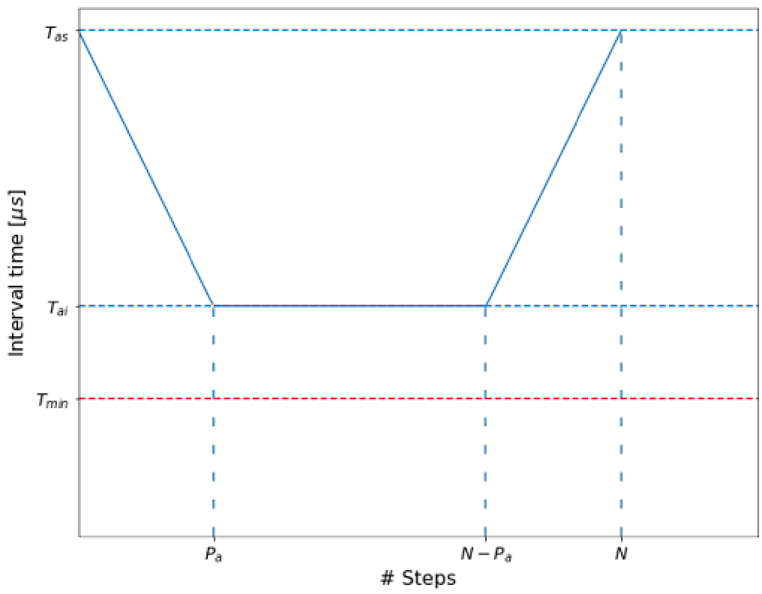

Fig. 3: Acceleration/Deceleration exmaple

To apply a change to any of these parameters, the **Apply** button needs to be used. This will send a message to the Portenta with for following structure: "p-6400-4800-40-2-200-4800-40-8", where the first four numerical values correspond to the azimuth parameters (resolution and acceleration), and the last 4 values correspond to the elevation parameters (except for the elevation resolution, by design choice, these parameters are currently not displayed on the GUI, but if needed they could be easily implemented). Note that there is always a default value already set in the Portenta for all of these parameters. Also, if no acceleration/deceleration routine is required, the aforementioned parameters can simply be set to zero.

3.3 Independent Movements

An option for independent movements both in azimuth and elevation is provided to move the antenna to an arbitrary position independent of the measurement routine. The leftmost radio button with a protractor icon is used to move each stepper motor to an *absolute* position, relative to the global reference set by software in the microcontroller. The **Reset angle** button sets the current position as angle zero. The remaining two radio buttons (counterclockwise/clockwise for azimuth and up/down for elevation) are designed to move the stepper motor an arbitrary angle independently of the global resolution set.

The string sent to the Portenta board has the structure x000, where x is a letter that determines whether an absolute, counterclockwise or clockwise movement is commanded in azimuth or elevation. The number 000 represents the number of steps dependent on the angle set at the azimuth/elevation spin box. An angle/step conversion is computed before sending the string according to the angular resolution set. For instance, if the counterclockwise radio button is checked, 360º is put at the Angle spin box, azimuth resolution is set at 200 step/turn, and the azimuth **Move** button is clicked, then `l200` is the message sent to the microcontroller, corresponding to a counterclockwise (leftward) movement. The remaining characters for the other radio buttons are `a` for absolute azimuth movements, `r` for clockwise (rightward) movements, `e` for absolute elevation movements, `u` for upward movements and `d` for downward movements.

3.4 Routine

Once the global angle reference is set, a measurement routine may be executed by clicking the **Start** button. The routine is designed so that a minimum of -180º and a maximum of 180º can be chosen for azimuth movements, and a minimum of -30º and a maximum of 60º for elevation movements.

A string with the format "`n-aIII-(l/r)FFF-eIII-eFFF-N`" is sent to the microcontroller. The first character `n` lets the microcontroller know that the following information is related to the routine. This is the same idea with `p` for parameters and `c` for connection. The first two sections "`aIII-(l/r)FFF`" are related to the initial and final azimuth angle spin boxes. These use the same format as with independent movements, that is, `aIII` represents an absolute azimuth movement to an absolute step which depends on the resolution set, and (l/r)FFF represents a relative movement counterclockwise or clockwise in order to reach the final angle set at the Final angle spin box. Likewise, the following two sections `eIII-eFFF` represent an absolute elevation movement related to the angles set at the Initial and Final elevation angle spin boxes. The last section `N` corresponds to the Rotations per elevation position spin box, which is the number of repetitions of azimuth movements for each elevation angle.

At the microcontroller, the routine consists of the following:

1. **Initialisation:** The azimuth and elevation motors move to the absolute position given by their respective Initial angle spin boxes.

2. **Repetitions:** The number of elevation movements to go from `eIII` to `eFFF` is computed according to the current elevation resolution. This is independent of the number of rotations per elevation position.

3. **For cycle:** For each elevation position, the azimuth motor moves to the final position set at the Final angle spin box. During this movement, captured data at each step is stored in the microcontroller RAM for it to be sent when the movement is complete. Then, the motor returns to the initial position without storing data travelling the shortest angle. This pair of movements is repeated `N` times for each elevation position. When the initial and final azimuth angle are set to -180º and 180º respectively, the second movement does not occur since the motor is already at its initial position. In this case the rotation and data transmission is continuous over 360 degrees. The data transfer time is negligible in any case.

3.4.1 Data sending

As already mentioned, data is sent once for each complete azimuth movement. This is done with `Serial.write()` function, which receives a pointer to memory where data is allocated. Metadata which includes times taken at each step, final position, and direction of movement is sent at the end of each data array. All data is sent as integers, whose size is of 4 bytes. Since the maximum value of each data is going to use at most 2 bytes (the rightmost bytes), the two leftmost bytes are always zero. This is taken advantage of to define a 4 byte arrangement which lets know the PC all info has been correctly sent for each movement, by setting an arbitrary number of bits to one at the leftmost bytes.

3.5 Save/Open Settings

Values set at each spin box and combo box can be stored as a JSON file by clicking `File > Save Routine Settings`. By clicking `Open Routine Settings`, a JSON file can be read to set previously chosen parameters.

3.6 Save/Open Data

Clicking `File > New Save File` prompts the starting of saving data in a CSV file. An example of a saved file is shown in Table 1, where the data can be plotted by considering the direction of movement, azimuth angular resolution and last azimuth angle measured. For instance, if the last azimuth angle is 10º, the movement direction is counterclockwise (positive), azimuth resolution is 360 step/turn (1 deg/step), and the length of the data array is 11, then data can be plotted in the interval $[0, 10]$ by steps of 1 degree. Open CSV data does nothing at the moment, but its future intended purpose is to plot previously saved data.

Date	Time	RPM	Last azimuth angle (°)	Movement direction	Azimuth resolution (step/turn)	Current elevation angle (º)	Repetition number	Elevation resolution (step/turn)	Data		
07 December 2022	15:33:50	114.33	79.93125	counterclockwise	6400	329.9625	1	6400	72	66	...
07 December 2022	15:33:50	114.33	79.93125	counterclockwise	6400	329.9625	2	6400	68	66	...
07 December 2022	15:33:51	114.33	79.93125	counterclockwise	6400	330.01875	1	6400	71	72	...
07 December 2022	15:33:51	114.33	79.93125	counterclockwise	6400	330.01875	2	6400	67	67	...
07 December 2022	15:33:52	114.33	79.93125	counterclockwise	6400	330.075	1	6400	68	63	...
07 December 2022	15:33:52	114.33	79.93125	counterclockwise	6400	330.075	2	6400	71	66	...
07 December 2022	15:33:53	114.33	79.93125	counterclockwise	6400	330.13125	1	6400	68	69	...
07 December 2022	15:33:53	114.33	79.93125	counterclockwise	6400	330.13125	2	6400	62	64	...
07 December 2022	15:33:54	114.33	79.93125	counterclockwise	6400	330.1875	1	6400	63	65	...

Table 1: CSV file example

Below the plot area, four labels are added which display characteristics of the measurements as well as the motor angular speed. From left to right, these are Peak Power, Peak Angle, Mean Power and Revolutions Per Minute.

Table 2 shows a comparison of angular velocities in RPM for different resolutions and acceleration parameters. Note that the maximum possible velocity reached at 6400 step/turn is 142.0 RPM without added delays. Delays are added for robustness, to prevent a drift at the absolute angular reference caused at the stepper motor in the development end.

Resolution (step/turn)	Accel. Period (rev)	Sup. Accel. Time (μs)	Inf. Accel. Time (μs)	RPM
6400	0.75	40	2	101.9
6400	0	0	0	142.0
3200	0.75	40	4	125.0
1600	0.75	40	8	137.9
800	0.75	40	8	146.5
400	0.75	40	8	153.7
200	0.75	40	8	156.2

Table 2: Angular velocities comparison.

4 LAN Remote Access to Stored Data

A PC running Windows 10 or 11 can be accessed remotely via LAN by another computer running any operative system natively. Another third party option is AnyDesk, which works on Windows, Linux and MacOS.

5 Adaptability to Resolution Scale Factor

As previously stated, all resolutions are emulated by skipping steps by a factor of 2^n over a maximum resolution, which in this case is 6400 step/turn. In case a different maximum resolution is used, both the Arduino and Python codes must be adapted.

In line 18 of `motorSerial.ino`, a global constant variable is defined as `N_rev_max = 6400`. This must be changed to whatever resolution is set as the maximum resolution. The number of steps to skip is defined as the division of the maximum resolution over the current resolution (see line 72 inside `forward` function).

In the GUI, new resolutions must be defined for the azimuth and elevation resolution combo boxes. In line 58 of `config.json` the list of angular resolutions are defined. Note that these correspond to 200, 400, 800, 1600, 3200, and 6400 step/turn respectively.