Underwater Acoustics Acquisition Sys Architecture and Design Rev 2

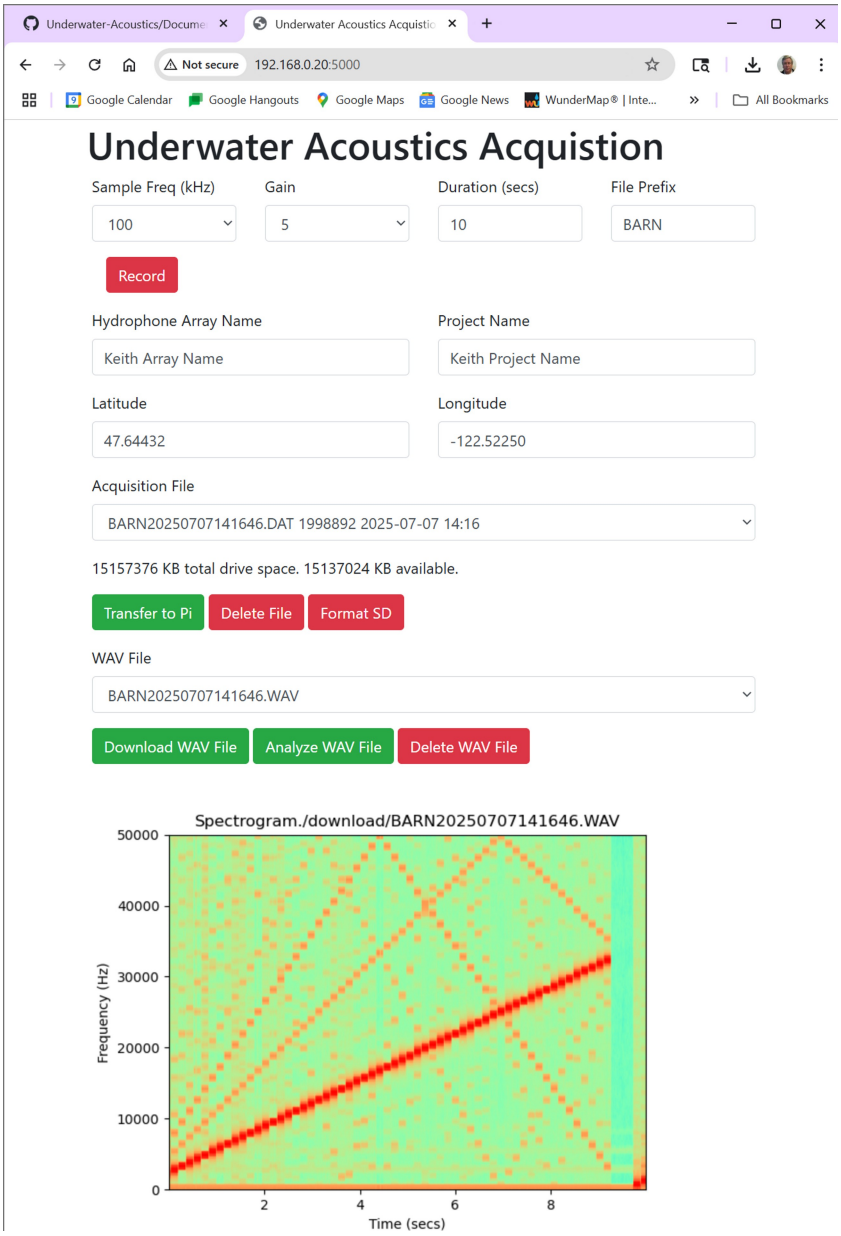
Sunday, October 1, 2023 1:38 PM

The Underwater Acoustics Acquisition System is based on an STM32H7x board that is responsible for the hydrophone analog signal acquisition and storing the recorded values on an SD card on the STM board.
The STM board is under the control of a Raspberry Pi that has a GUI which the operator uses to control the acquisition

The Raspberry Pi communicates commands to the STM32 using an SPI interface. The STM32 executes the requested command and, if required, returns data to the Raspberry Pi.

The STM32 board also has a serial port through which diagnostics messages are sent during normal operation

The User Interface looks like this and is served up by the Raspberry Pi using a Flask Server



The user defines the parameters for the upcoming recording including duration, sampling frequency and gain.
The user can select from 7 different gain levels
The filename of the acquired data is the date and time with a file prefix chosen from the user interfac

When initially started, the STM32 mounts the SD. If the SD card cannot be mounted, a error message is sent to the STM32 diagnostics port

When the **Record** button is pressed, all parameters for the recording are sent from the Raspberry Pi to the STM32.
The Raspberry Pi also sends the latest date/time to the STM32 which the STM32 uses to set it's internal real time clock
The STM sets up the ADC, DMA and initiates data collection.
ADC data is captured using the 16 bit ADC on the STM32.
ADC data is captured continually with the completion of each ADC conversion initiating the next ADC conversion
Captured data is stored in memory that is organized as 16k samples. The memory is further subdivided into two halves (an upper and lower buffer). Initially, data is saved to the lower buffer and when the buffer is full, further data is saved to the upper buffer. At this time, the STM32 initiates the use of DMA to transfer the contents of the lower buffer to the SD card. This flip flop process continues until the end of the Duration.
At the end of the duration the STM32 closes the file and increments the "highest file number" counter
The STM32 also sets the date/time on the file

When the **Transfer to Pi** button is pressed, the Acquisition File selected from the dropdown list is moved from the STM32 SD Card to the Raspberry Pi. Data is transferred in large blocks that are sequentially requested by the Raspberry Pi which is always the Master of the SPI interface

The DAT file format on the STM32 SD card is essentially a standard WAV file format with the exception that the analog values are stored as 16 bit unsigned integers. When the data is sent from the STM32 to the Raspberry Pi, the STM32 subtracts 0x8000 from the analog values to provide a file that has data values stored as 16 bit signed integers.
The Raspberry Pi opens the file in the file system
The Raspberry Pi is the Master for the SPI interface and to ensure it does not outpace the STM32, a BUSY handshake line is implemented such that the STM32 will tell the Raspberry Pi when it is busy and cannot accept a further SPI request. The Raspberry Pi only sends data requests when the STM32 is not busy. The BUSY line is implemented on PF3-D10 on the STM32. The BUSY line connects to GPIO 27 on the Raspberry Pi

The communication protocol between the STM and Raspberry Pi is shown in the PPT

Under normal operation the STM is waiting from commands from the Raspberry Pi. When the Raspberry Pi sends a command, it first takes the Chip Select low (SPI_NCS)

Analog signals received by the STM32 ADC must be between 0.5v and 3.0v

GPS Communications

Field	NMEA_buff	Structure	Description	Symbol	Example
1	0	$GPGGA	Log header		$GPGGA
2	1	utc	UTC time status of position (hours/minutes/seconds/ decimal seconds)	hhmmss.ss	202134
3	2	lat	Latitude (DDmm.mm)	llll.ll	5106.9847
4	3	lat dir	Latitude direction (N = North, S = South)	a	N
5	4	lon	Longitude (DDDmm.mm)	yyyyy.yy	11402.2986
6	5	lon dir	Longitude direction (E = East, W = West)	a	W
7	6	quality	refer to Table: GPS Quality Indicators	x	1
8	7	# sats	Number of satellites in use. May be different to the number in view	xx	10
9	8	hdop	Horizontal dilution of precision	x.x	1
10	9	alt	Antenna altitude above/below mean sea level	x.x	1062.22
11	10	a-units	Units of antenna altitude (M = metres)	M	M
12	11	undulation	Undulation - the relationship between the geoid and the WGS84 ellipsoid	x.x	-16.271
13	12	u-units	Units of undulation (M = metres)	M	M
14	13	age	Age of correction data (in seconds)	xx	(empty when no differential data is present)
15	14	stn ID	Differential base station ID	xxxx	(empty when no differential data is present)
16	15	*xx	Check sum	*hh	*48
17	16	[CR][LF]	Sentence terminator		[CR][LF]

GPS Quality Indicators	
Indicator	Description
0	Fix not available or invalid
1	Single point
	Converging PPP (TerraStar-L)
2	Pseudorange differential
	Converged PPP (TerraStar-L)
	Converging PPP (TerraStar-C, TerraStar-C PRO, TerraStar-X)
4	RTK fixed ambiguity solution
5	RTK floating ambiguity solution
	Converged PPP (TerraStar-C, TerraStar-C PRO, TerraStar-X)
6	Dead reckoning mode
7	Manual input mode (fixed position)
8	Simulator mode
9	WAAS (SBAS)1