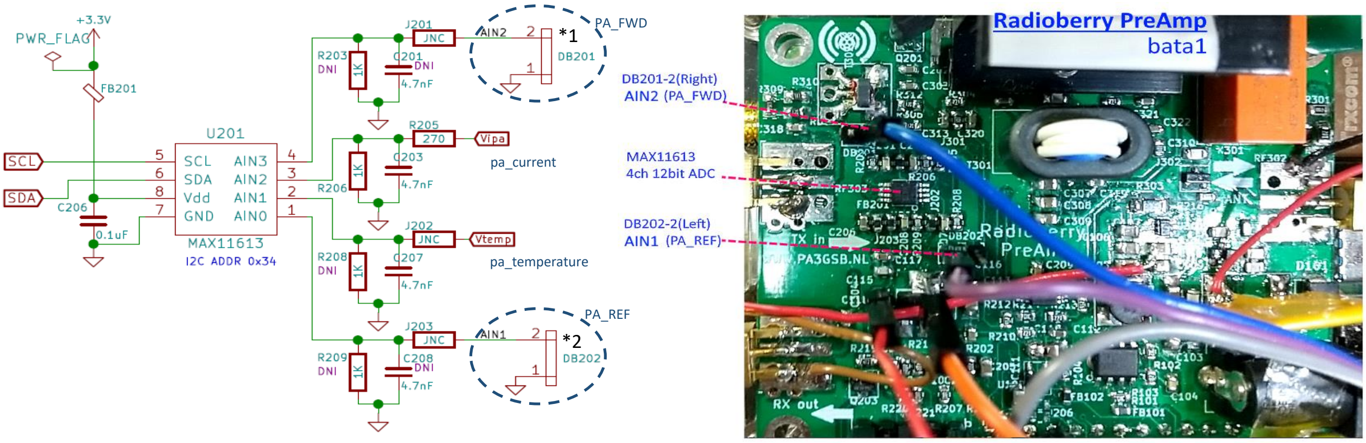
Try adding RF-Power measurement to the Radioberry PreAmp

1. Around the ADC inputs.

https://github.com/pa3gsb/Radioberry-2.x/wiki/Radioberry-preamp



Note1: AIN2(PA_FWD) <- 1.61V (AA battery)
Note2: AIN1(PA_REF) <- 0.73V (Divided voltage)

2. firmware modify.

https://github.com/pa3gsb/Radioberry-2.x/tree/master/SBC/rpi-4/device_driver/firmware

The packet part was added with reference to the protocol. The measurement part was added by imitating the temperature measurement part.

radioberry.c (Added red text sections)

```
void fillPacketToSend() {
    memset(hpsdrdata,0,1032);
 :
        if ( last_sequence_number % 2 == 0) {
            // if i2c_measure_module_active; the temperature of module is used, otherwise the RPI temp.
            hpsdrdata[11 + coarse_pointer] = 0x08 | (rb_control & 0x07);   // C0
            if (i2c_measure_module_active) {
                hpsdrdata[12 + coarse_pointer] = ((pa_temp >> 8) & 0xFF);  // C1
                hpsdrdata[13 + coarse_pointer] = (pa_temp & 0xFF);         // C2
                hpsdrdata[14 + coarse_pointer] = ((pa_fwd >> 8) & 0xFF);   // C3
                hpsdrdata[15 + coarse_pointer] = (pa_fwd & 0xFF);          // C4
            } else {
                hpsdrdata[12 + coarse_pointer] = ((sys_temp >> 8) & 0xFF);
                hpsdrdata[13 + coarse_pointer] = (sys_temp & 0xFF);
            }
        } else {
            hpsdrdata[11 + coarse_pointer] = 0x10 | (rb_control & 0x07);  // C0
            hpsdrdata[12 + coarse_pointer] = ((pa_ref >> 8) & 0xFF);      // C1
            hpsdrdata[13 + coarse_pointer] = (pa_ref & 0xFF);             // C2
            hpsdrdata[14 + coarse_pointer] = ((pa_current >> 8) & 0xFF);  // C3
            hpsdrdata[15 + coarse_pointer] = (pa_current & 0xFF);         // C4
        }
    }
}
 :
static void *rb_measure_thread(void *arg) {
    // temperature == (((T*.01)+.5)/3.26)*4096   if pa temperature > 50C (=1256) switch pa off! (pa_temp_ok)
    int measured_temp_ok_count = 0;
    while(1) {
        sem_wait(&i2c_meas);
        if (i2c_measure_module_active){
        read_I2C_measure(&pa_current, &pa_temp, &pa_fwd, &pa_ref);
    //  fprintf(stderr, "PA_FWD is %d  PA_REF is %d ¥n", pa_fwd, pa_ref);
        }
 :
}
```

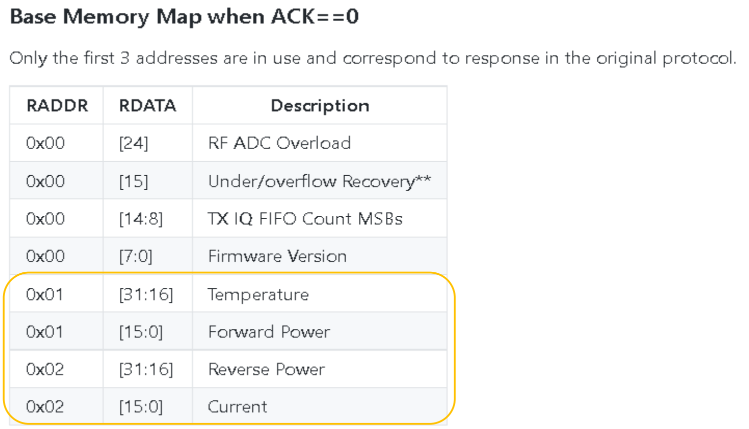
measure.c

```
void read_I2C_measure(int *current, int *temperature, int *fwd, int *ref){

    uint8_t measure_data[8] ={0};

    int result = read(fd_i2c_measure, measure_data, 8);

    *temperature = (int)(((measure_data[2] & 0x0F) <<8) | measure_data[3]);
    *current = (int)(((measure_data[4] & 0x0F) <<8) | measure_data[5]);
    *ref = (int)(((measure_data[0] & 0x0F) <<8) | measure_data[1]);
    *fwd = (int)(((measure_data[6] & 0x0F) <<8) | measure_data[7]);
};
```

* Also change the prototype declaration of this function in 'measure.h'.

The ADC data to RPi4 Protocol.

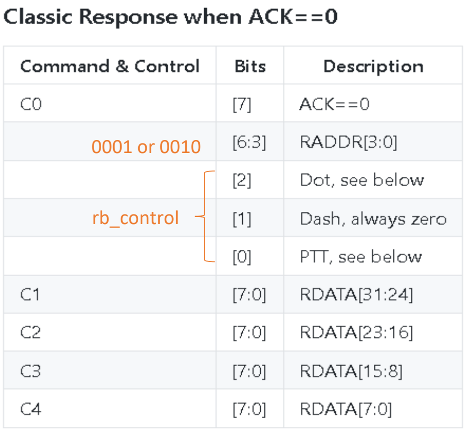
// inform the SDR about the radioberry control status.
// https://github.com/softerhardware/Hermes-Lite2/wiki/Protocol



Base Memory Map when ACK==0

Only the first 3 addresses are in use and correspond to response in the original protocol.

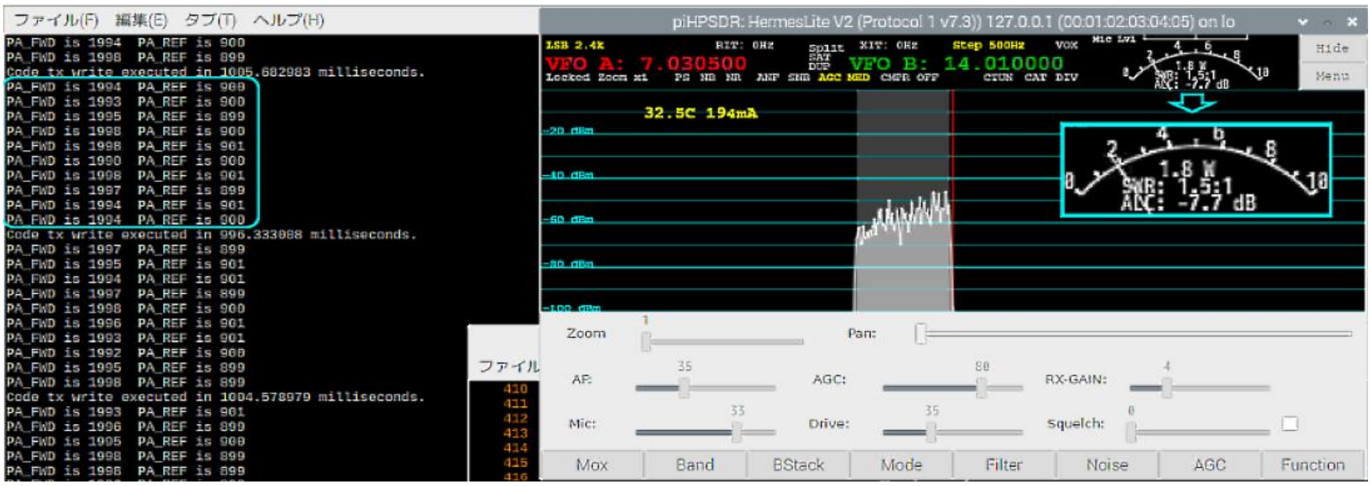
RADDR	RDATA	Description
0x00	[24]	RF ADC Overload
0x00	[15]	Under/overflow Recovery**
0x00	[14:8]	TX IQ FIFO Count MSBs
0x00	[7:0]	Firmware Version
0x01	[31:16]	Temperature
0x01	[15:0]	Forward Power
0x02	[31:16]	Reverse Power
0x02	[15:0]	Current



Classic Response when ACK==0

Command & Control	Bits	Description
C0	[7]	ACK==0
0001 or 0010	[6:3]	RADDR[3:0]
	[2]	Dot, see below
rb_control	[1]	Dash, always zero
	[0]	PTT, see below
C1	[7:0]	RDATA[31:24]
C2	[7:0]	RDATA[23:16]
C3	[7:0]	RDATA[15:8]
C4	[7:0]	RDATA[7:0]

3. Experiment results. (piHPSDR)



```
PA_FWD is 1994  PA_REF is 900
PA_FWD is 1998  PA_REF is 899
Code tx write executed in 1005.682983 milliseconds.
PA_FWD is 1994  PA_REF is 909
PA_FWD is 1993  PA_REF is 900
PA_FWD is 1995  PA_REF is 899
PA_FWD is 1998  PA_REF is 900
PA_FWD is 1998  PA_REF is 901
PA_FWD is 1990  PA_REF is 900
PA_FWD is 1998  PA_REF is 901
PA_FWD is 1997  PA_REF is 899
PA_FWD is 1994  PA_REF is 901
PA_FWD is 1994  PA_REF is 900
Code tx write executed in 996.333088 milliseconds.
PA_FWD is 1997  PA_REF is 899
PA_FWD is 1995  PA_REF is 901
PA_FWD is 1994  PA_REF is 901
PA_FWD is 1997  PA_REF is 899
PA_FWD is 1998  PA_REF is 900
PA_FWD is 1996  PA_REF is 901
PA_FWD is 1993  PA_REF is 901
PA_FWD is 1992  PA_REF is 908
PA_FWD is 1995  PA_REF is 899
PA_FWD is 1998  PA_REF is 899
Code tx write executed in 1004.578979 milliseconds.
PA_FWD is 1993  PA_REF is 901
PA_FWD is 1996  PA_REF is 899
PA_FWD is 1995  PA_REF is 909
PA_FWD is 1998  PA_REF is 899
PA_FWD is 1996  PA_REF is 899
```

↑ PA_FWD is abt 1997 : 1997/4095*3.3V=1.61V (OK)
↑ PA_REF is abt 900 : 900/4095*3.3V=0.73V (OK)

↑ Power shows 1.8W (@Vfwd=1.61V)
SWR is 1.5 ＊

piHPSDR displays:
Pow=1.8W SWR=1.5 when Pre-Amp AIN2,1 : Vfwd=1.61V, Vref=0.73V
Pow=1.8W SWR=2.0 when Pre-Amp AIN2,1 : Vfwd=1.61V, Vref=0.92V

* VSWR = Vmax/Vmin = (|Vfwd|+|Vref|) / (|Vfwd|-|Vref|)
> VSWR=2.66 @simply Vref=1.61, Vref=0.73...
> maybe between power and voltage is not linear , Especially QRPp .
> so, must be correcting the readings

Need to get data of 'piHPSDR_Power vs. AIN' and 'Vfwd/Vref vs. Power'

18 July 2021 jg1twp