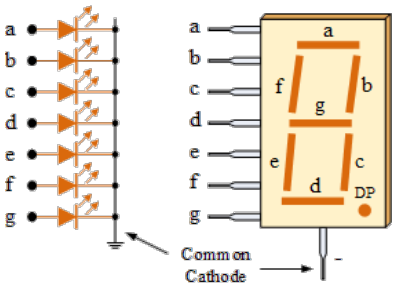
Introduction

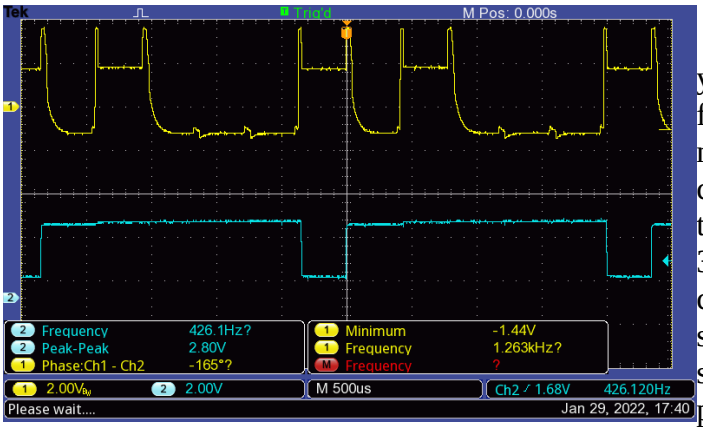
At the 80’ years arrive to the market the firsts Yaesu transceivers with frequencies digital memory what allowed to fix the frequency at memorized value. On example was the classic transceiver FT-107m that came equipped with the DMS unit, but it did not has a computer cable to control it. In order to fill this gap, this article will describe the develpment and construction of a interface to control DMS unit form an yaesu FT-107M through a computer letting more easy to use it on digital modes. The project is composed by a Arduino Nano circuit and the source code that let us to control frequency and the radio ptt.

The hurdle and the overcoming

The problem of reading 7 segments display with an Arduino [2] was discussed with the indicating that a part of the solution was related to implementation of interrupt processes [1] since the routine should be executed at time that event occur, that is, when the display is turned on.

A 7 segment display are compound from seven led plus a digit named DP, decimal point. The picture presents an example, at common on cathode configuration found at counter unit from FT-107M. Each on of the six 7 segment display receive a start “On” signal at its cathodes as same time that the a to g signals (plus DO when necessary) are placed at state on/off. The start “On” signal occurs when the blue trace falls and remains “On” until the blue trace rises. Here, we see an example of event that must be acquired through a interrupt processes to be analysed correctaly.





The scope image present two signals, the yellow cannel (batman shape) was measured from “g” segment and the blue cannel was measured from “d1” pin that represent the catodo from first display. What the picture tell us is that when signal ‘d1’ falls from 3.5V to 1V, d1 display is start “On” and conduct current. At this moment, the yellow signal rise and “g” segmet is state ON. The signals have a frequency around 426hz, or period T 2.35 ms. The gap have a time width of 390us and this is the time that the Arduino have to capture the 7 segments (a - g) during a interrupt processes time interval. We can see the “g” segment ON for display d3 and OFF for the anothers.
From the point of view of software, the “d1” display signal is acquired by PIN 4 (vide schematic circuit)

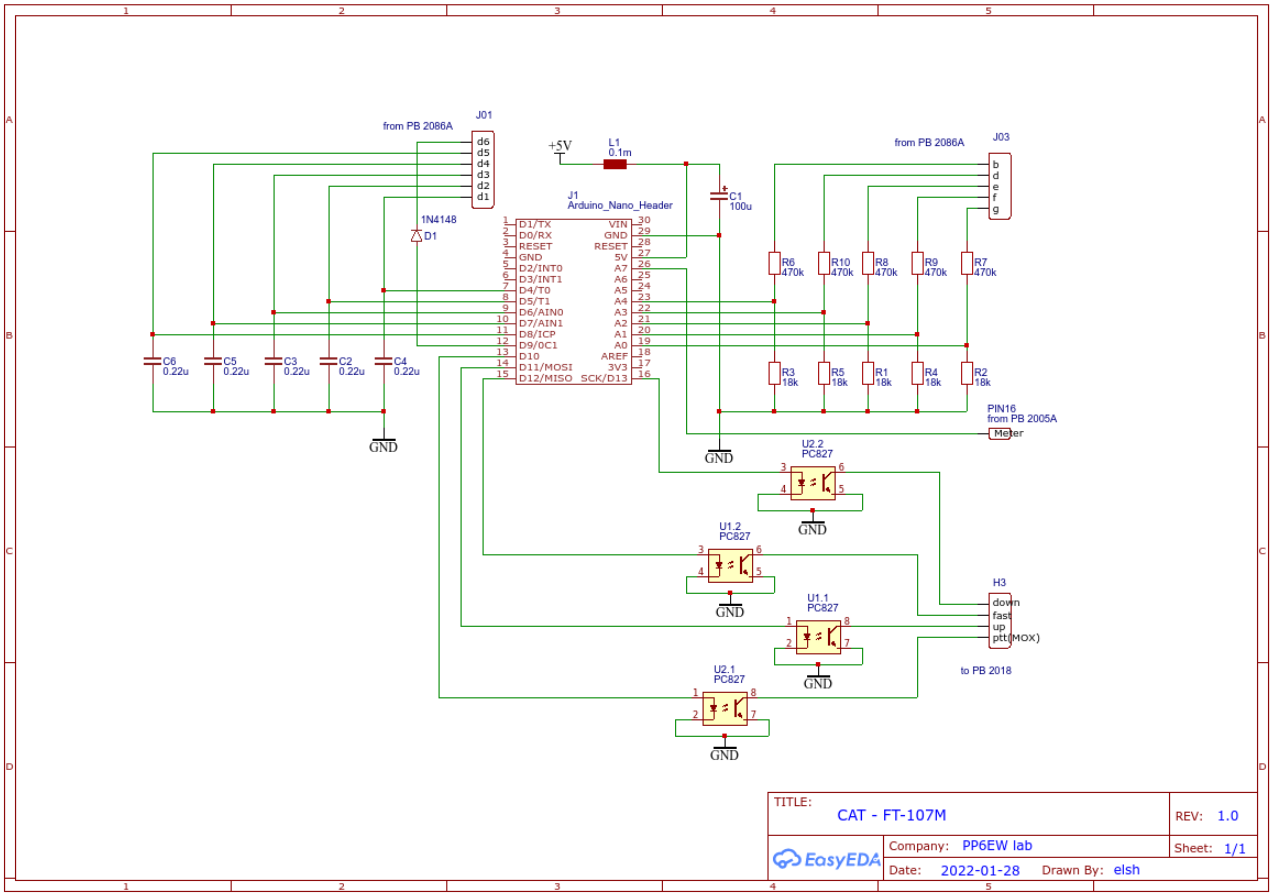
```
pinMode(PIN4, INPUT);
enableInterrupt(PIN4 | PINCHANGEINTERRUPT, interruptFunction_d1, FALLING);
where procedure  interruptFunction_d1 :
void interruptFunction_d1() {
  for (int i = 0; i < 5; i++) { //  bdefg
    if ( (analogRead(analog_pins[i])) > 90){    // referent to yellow signal
  dial[0][i]=1;
  }else dial[0][i]=0;
}
```

read 5 analog_pins and update the 5 component vector dial[0][i] with a sequence that represent a digit number readed.

Another hamper are related to sample speed at the analog channels that was, by default, inssuficient to complete the data acquisition at a time interval of 390us. The solution was to make the rate of sample [3-4] more fast. From setup(), a code segment
sbi(ADCSRA,ADPS2) ;
cbi(ADCSRA,ADPS1) ;
cbi(ADCSRA,ADPS0) ;
change sample rate from 9600 Hz to 76.8Khz.

Circuit description

The circuit can be seen at fig 1.The Arduino Nano[7] was choiced because it is a board with low cost and make the design compact and as simple as possible.



From the Arduino Nano board we use 6 digital input (D4-D9) and 5 analog inputs (A0-A4) to reads the multiplexed frequency from PB 2086A. The S-meter signal is read at A7 input and 4 digital output are used to drive 4 photocoupler ports to control up, down, fast frequencies shifts and the PTT tx control through the pb 2018 board.
The PB 2086A is the display control unit. In each catodo display there are a transistor working on cutoff and saturation. When we measure signals (d1-d6) we obtain 3.5 V high level receive 3.5V and low level around 1 V that can be used as high and low switch by arduino input digital ports [5]. The 5 capacitors plugged to d1-d5 inputs modify the original signal shape and create the blue signal presented above. When “d6” display is ON, the diode 1N4148 start to conduct and D9 input is setted in HIGH mode.

The anode signal comes from transistors that working as switches for the 7 segments. An example for “g” segment is the yellow signal that is ON when the value is above a threshold. Was not necessary to read 7 segments because it has redundant information in it, so only 5 segments was choiced to represents the numbers readed without lost data. The A7 analog input receive a signal from 0 - 0.25V and the best way to read low voltage is to redefine analogReference.

Software

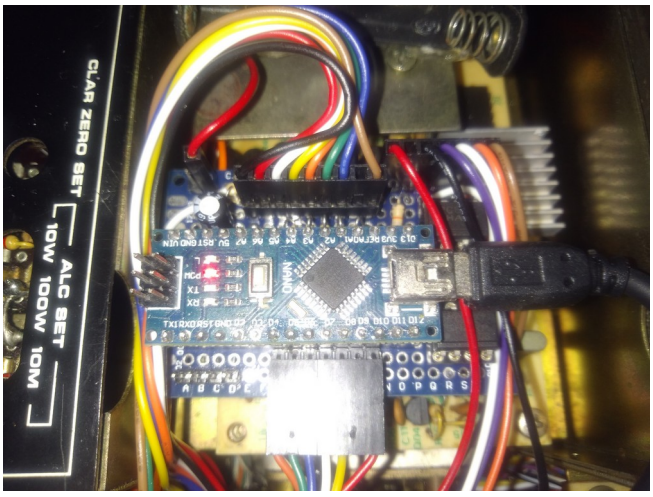
In the section setup() we found
analogReference(INTERNAL); // AREF 1.1V
what change the analogReference [6] to INTERNAl.
As same time we found the change on the sample rate at analog channels and interruptFunction described above.

To allows for much faster I/O control [8] we use
DDRC = B00000000; // set all analog pins as input

To make the CAT interface completely usable from a ham radio professional point of view, we place the source code inside the codes developed to act as an Yaesu radio [9].

Construction

The prototype was constructed over a PCB 14x20 holes. It was possible to plug PCB over the pb 2086A , making it a very compact .



Modifications, adjusts and tests

In this section will be see how to fit the project to yours needs. There are two type of possible modifications to do: mods at hardware and at the software. The resistor’s network compose by R1 to R10 create a voltage dividers that should adjust by software setting when we not used 1% resistors. Other possible modification are related to change the value of capacitors C2 to C6 or may be take off its. May be work without these capacitors.

To test the circuit I sugest to use flrig software. Of course that you can start using a source code that let you to read the frequency and see the output through the serial port.
https://github.com/bunhoel/FT107m/blob/main/input_analog_5segments_7d_test_ok_serial1.ino

Reference

1) https://github.com/GreyGnome/EnableInterrupt

2) https://arduino.stackexchange.com/questions/3731/reading-a-10-pin-7-segment-2-digits-sing-arduino-1-please-help
3) Howto make an Arduino fast enough to…
http://www.optiloading.be/willem/Arduino/speeding.pdf
4) https://forum.arduino.cc/t/faster-analog-read/6604/4
5) https://www.arduino.cc/reference/en/language/variables/constants/constants/
6) https://www.arduino.cc/reference/en/language/functions/analog-io/analogreference/
7) https://www.arduino.cc/en/pmwiki.php?n=Main/ArduinoBoardNano
8) https://www.instructables.com/Arduino-and-Port-Manipulation/
9) https://github.com/pavelmc/FT857d/blob/master/src/ft857d.cpp