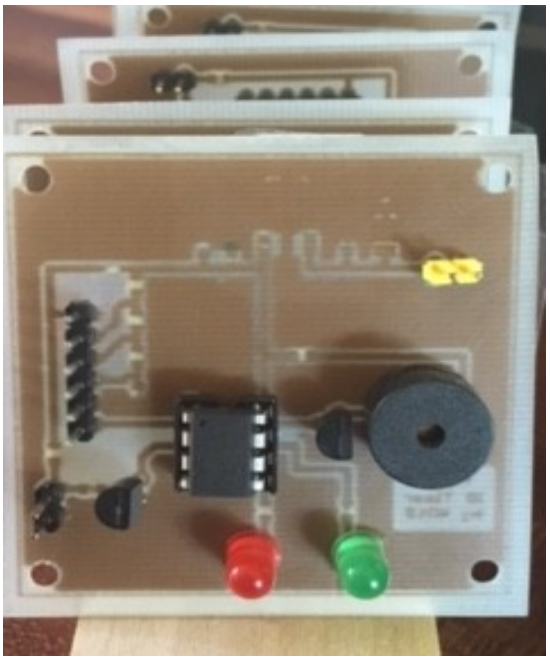
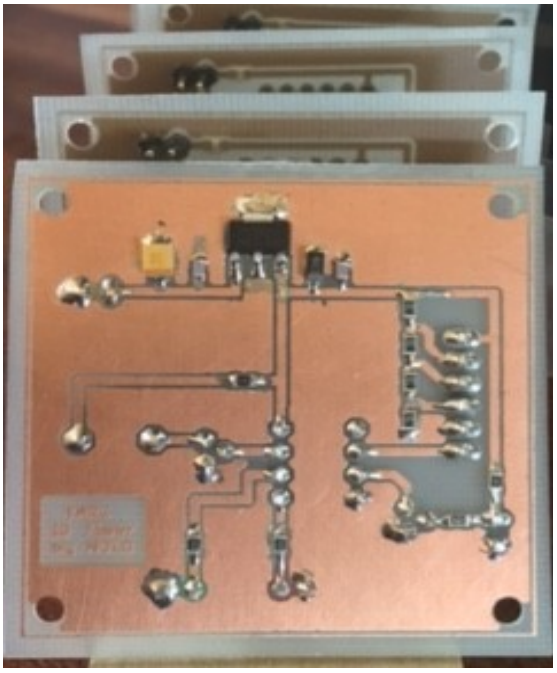
YARC ID Timer Project 20160404

Introduction

During the presentation on PCB Prototyping at the April 2016 YARC meeting I presented a circuit of a simple ID Timer (IDT). This document details the circuit, parts needed for its assembly as well as a few operational and design notes.





Description

The IDT is a simplistic timer offering, 2M, 3M, 5M, 10M and 15M timeout measurements. The board was designed as an example circuit during the April 2016 YARC talk on PCB prototyping and uses both thru-hole and SMT components. Don WB7TPH suggested a Ham ID Timer circuit; so taking his idea to the next level and utilizing an ATtiny85 microprocessor that I had never used before, this timer circuit was born.

The IDT utilizes an ATtiny85 8-pin Arduino/AVR compatible microprocessor executing software which tracks inputs, controls outputs and performs the timer functions. This small 8-pin chip has 8-kbytes of storage of which 25% used for this project. Software[1] for the ATTiny85 was written in 'C' language using the Arduino sketch utility.

1 Software can be downloaded from: https://github.com/dtheriault/YARC.git

YARC ID Timer Project 20160404

The IDT has 3 outputs;

Green LED – Indicates start of timer operation and which begins at ativation of PTT input pin

Red LED – Indicates timer expiration and time to release the PTT to reset timer and alarm

Piezo Buzzer – Audible alarm when a timer has expired and continues until release of PTT

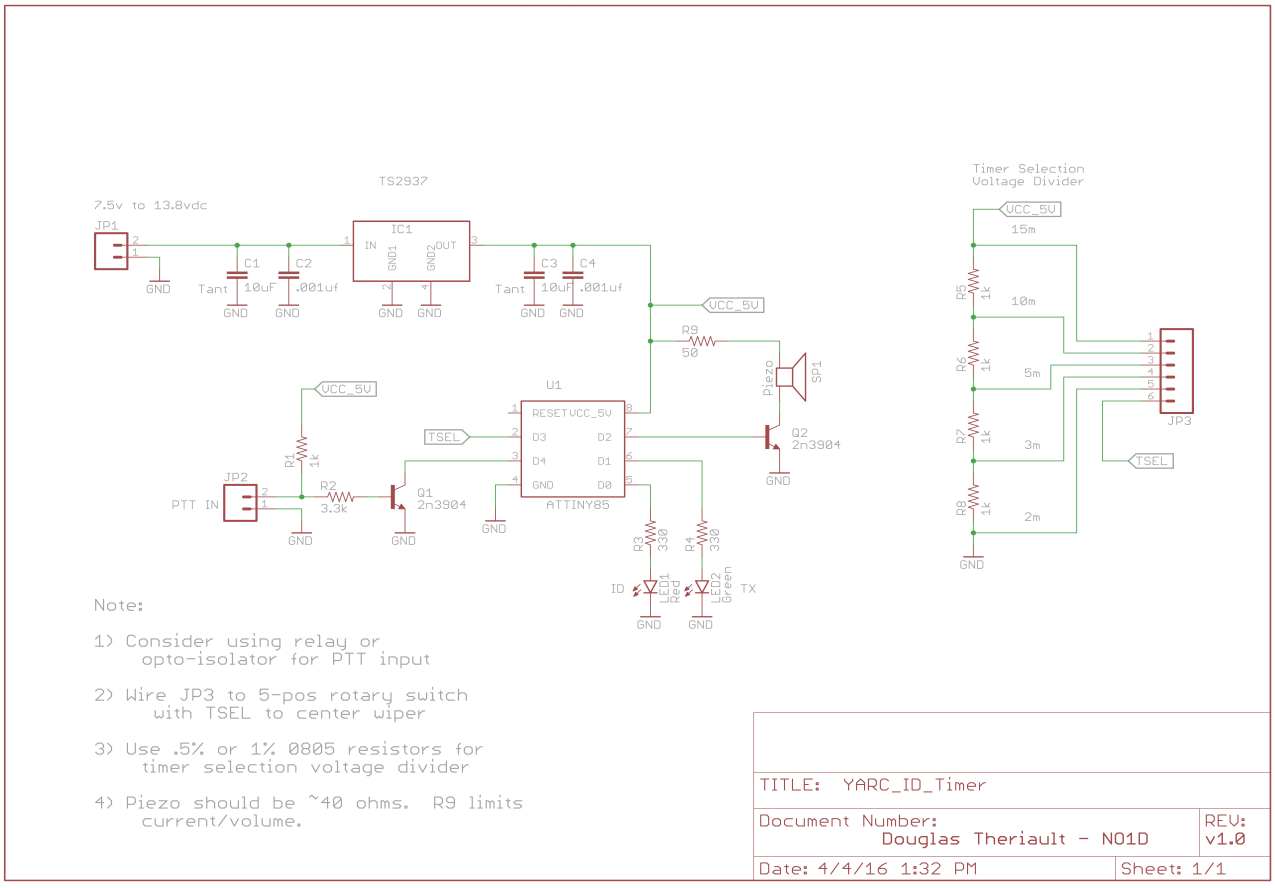
The IDT has 3 inputs:

PTT – JP2, which can be wired to the PTT output from a transceiver/transmitter, or to a manual switch or relay. This pin is pulled up via 1k resistor to VCC (+5v). Activates timer when brought low.

Timer Selection – JP3, A (6) pin header[2] connects to rotary switch allowing selection of timeouts.

VCC – JP1, The board requires voltage input of 7.5 vdc to 13.8vdc for proper operation via JP1. An on board 5vdc regulator assures stable operation from various input voltages. The circuit can operate from a 9vdc battery.

Schematic



2 The blank PCB's offered at YARC had 5-pin headers for JP3 and require a wire to +5v in order to select the 15m timer

YARC ID Timer Project 20160404

Power Supply

The IDT board uses a simple 78L05 linear voltage regulator to provide a stable 5v supply to the ATtiny85 microprocessor chip. The package is SMD will provide up to 100ma current. Basic SMD filtering caps are placed around the 78L05.

Input supplied voltage should be between 7.5vdc to 13.8vdc. No reverse protection diode is present so be careful not to reverse power supply leads at JP1.

JP1 should connect thru an on/off switch to prevent battery drain if if one is used as a supply. A 9vdc battery was used for the demo.

Piezo Alarm

Pin 7 of the ATTiny85 feeds a 2N3904 buffer which drives a 41Ohm Piezo alarm. Piezo buzzer in turn connects directly to the 5v supply thru a 50 ohm resistor which limits current passed by the 2N3904 buffer and provides reasonable sound level. 50 ohm resister may be shorted for max volume.

The ATTiny85 generates a 50% duty cycle square wave at approx 2000 hz for peak alarm sound. Upon expiration of a timer, the Piezo will sound at 1s intervals until PTT has been released.

PTT

The PTT input pin 2 at header JP2 is pulled up to 5v via a 1k resistor allowing control of the timer manually using a toggle switch. Timer begins counting when the PTT input is brought low, and resets when returns back high.

Connection to a transceiver/transmitter so the timer automatically begins when transmission starts may require use of an external relay or opto-isolator chip, driven by the PTT output from a rig. Since PTT outputs vary from rig-to-rig, rig interfacing effort is left for an exercise by the operator.

The PTT input is active low, requiring the PTT to be pulled to the GND state in order to drive the 2N3904 buffer interfacing to the ATtiny85 Pin 3 (D4).

NOTE: *Recommend using either a relay or opto-isolator circuit to toggle PTT when interfacing to a rig.*

Timer Selection – Voltage Divider

To provide a selection of timing intervals rather than a single fixed 10m timer, a remaining pin on the ATtiny85 was configured as an ADC input to read a voltage from a simple voltage divider network. The voltage divider uses .5% 1k resistors (although 1% could be used also) to provide accurate reference voltage for each of the timer selection options. Resistors >1% may require software changes for proper timer selection.

YARC ID Timer Project 20160404

Timer Selection	Voltage
2 Minutes	0 vdc
3 Minutes	1.25 vdc
5 Minutes	2.5 vdc
10 Minutes	3.75 vdc
15 Minutes	5.00 vdc

The ATtiny85 ADC samples by default are 10bit values. The sample values have been scaled down to 4 bits simplifying measurement of the voltages from the divider network. This reduced the range checking in software needed to accommodate variances in voltage readings even with 1% resistors. Optimizations in software can be made by taking multiple samples and averaging results if higher granularity is desired.

It is important that an accurate reference voltage is provided in order for the software to select a valid timeout selection. If the IDT displays an alternating RED/GREEN LED and Piezo alarm sounds, this is an indication that the voltage measured is invalid and the voltage divider network should be checked.

Header JP3 should connect to a simple rotary switch for timer selection.

JP3 Pin	Switch	Selection
1	1	15 minute selection
2	2	10 minute selection
3	3	5 minute selection
4	4	3 minute selection
5	5	2 minute selection
6	6 - Common Wiper	TSEL input to ATtiny85

Timing Accuracy

The ATtiny85 is configured to use its internal 8Mhz clock. Atmel indicated clocking should be accurate to 1% as configured from the factory. I found most chips which averaged 1.4% to 1.8% fast using manual stopwatch method of measurement and many more off by 10%.

Timeout values in software were adjusted to bring the accuracy to an average of +/- .1% of the desired timeout. A calibration routine was written that allows the internal clock to be output on pin 5 (PB0) at a rate of 4Mhz. In addition there is a #define CALIBRATION value that can be tweaked to move the internal clock higher/lower in order to fine tune the internal clock to < 1% accuracy. Please note that the clock can drift based on temperature and VCC which was not measured. VCC should be stable

using the regulator so temperature would be the major factor affecting stability of the internal clock.

Further improvements (optimizations) can be made to the software in order to improve timer accuracy. Current software uses a very basic “polling” based algorithm however use of Arduino timers, timer calibration and interrupts higher accuracy can be achieved. This is left as an exercise for the amateur who wishes to pursue the Arduino/AVR architectures further.

Bill of Materials (BOM)

Identifier	Value	Package	Notes
R1	1k	0805	PTT pull-up
R2	3.3k	0805	PTT bias control
R3,R4	330	0805	Current limiter for LED
R5, R6, R7, R8	1k	0805	.5% or 1% tolerance recommended Voltage Divider network for Timer Select
R9	50	0805	Short for max volume
C1	10uF Tantalum	1210	Power input filter capacitor
C2	.001uF	0805	Power input filter capacitor
C3	10uF Tantalum	1206	Regulator output filter capacitor
C4	.001uF	0805	Regulator output filter capacitor
Q1, Q2	2N3904	Through-hole TO-92	Buffer amplifiers, 2n2222 use possible
IC1	TS2937	SOT-223	5v @500ma Ultra Low Dropout Regulator Mouser: 821-TS2937CW50
U1	ATtiny85	Through-hole DIP-8	8-pin DIP socket recommended to facilitate ease of re-programming microprocessor.
SP1	42 ohm Piezo	Through-hole	12mm Thru-Hole Piezo Speaker, 2.048khz Sparkfun: com-07950
LED1	Red	Through-hole	5mm size
LED2	Green	Through-hole	5mm size
JP1, JP2	2 pin header	Through-hole	.1mm spacing
JP3	6 pin header	Through-hole	.1mm spacing

YARC ID Timer Project 20160404

Operation & Construction Notes

The IDT executes a simple 3 state finite state machine in software. The following table outlines each state.

State	Description
IDLE	PTT input is inactive Red / Green LED's are extinguished Timeout counter is cleared Timer Selection (TSEL) voltage is read during main loop 100ms polling interval
TIMING	PTT is now active, timer begins Green LED is illuminated Timer selection is committed. Any changes to TSEL once timing starts is ignored If PTT goes inactive, before timer expiration, return to the IDLE state Increment timer count value, check for timeout condition every 100ms. Check if timeout is reached, if so, enter the ALARM state
ALARM	Activate the Red ID LED Pulse the Piezo Buzzer at approx 1s intervals When PTT is inactive, return to the IDLE state

Its important to note that the timer selection switch is only read during the IDLE state polling interval. Once the PTT has been activated and TIMING state entered, TSEL no longer is read so no changes to timeout value is allowed. You must release PTT to once again read the timer select switch and configure a new timeout setting.

Once the ALARM state has been activated, the only way to cancel the Alarm and the annoying buzzer is to release the PTT switch (after identification of course) or remove power to the IDT.

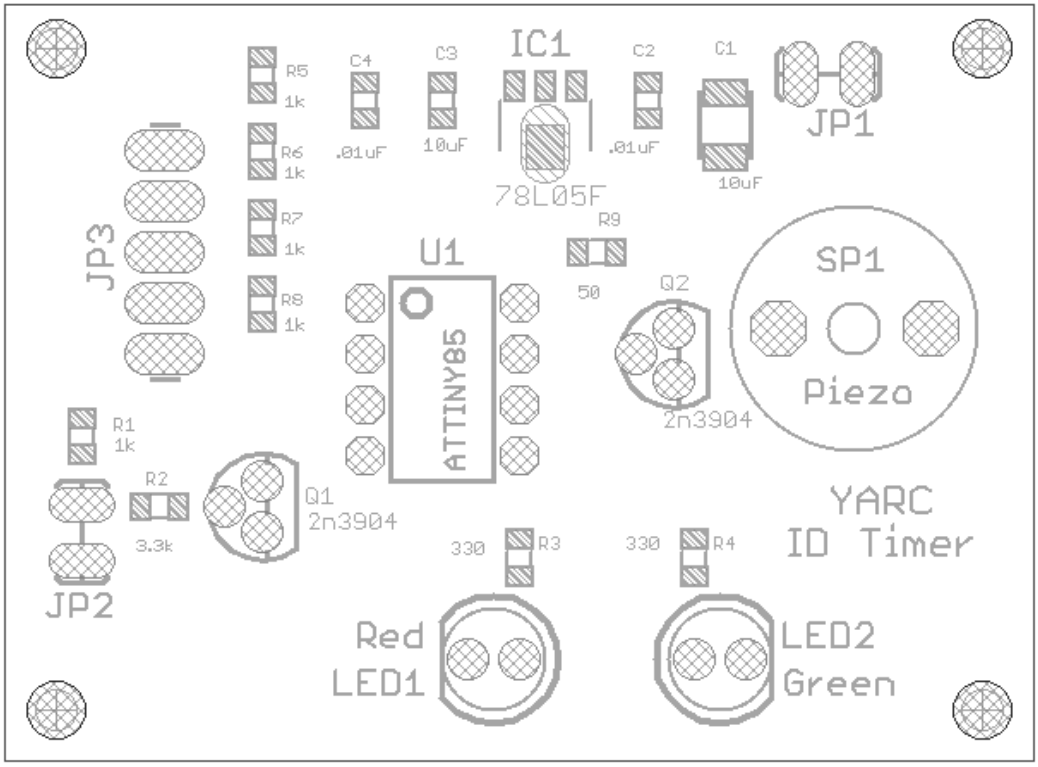
The board uses mix of through-hole and SMT devices for the prototyping talk. I have not had success yet at printing silk-screen layer using toner transfer or applying UV conformal coating. To ease construction, best to print out of the schematic, bottom and the silk screen layers and have handy to assist in parts placement and soldering them onto the board. Luckily this is a simple board with minimum number of parts and pretty straight forward as to parts placement.

The board is single sided one, with through-hole parts added to the non-copper side of the board. All SMT parts are soldered to the copper side which was printed as the bottom layer.

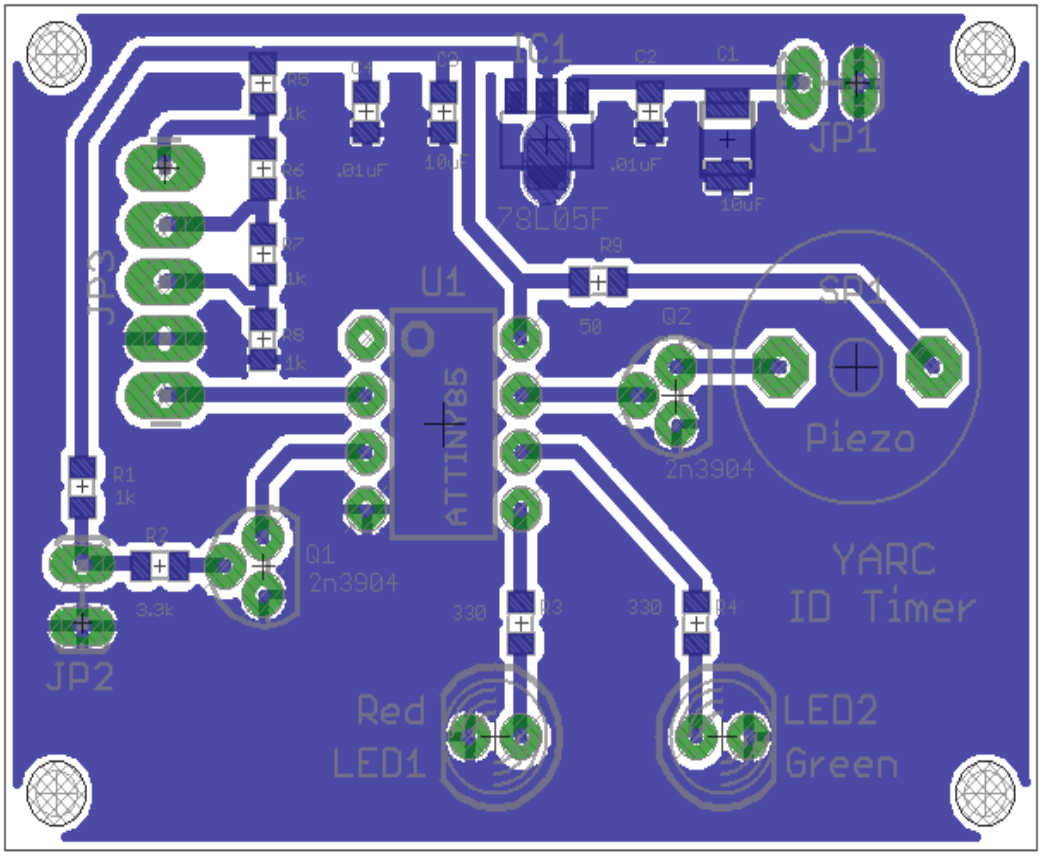
YARC ID Timer Project 20160404

Silk Screen and Bottom Layer

The following images are of the bottom layer traces and silk screen layer from the board files. They may be useful during assembly of the board. Please note these images are enlarged and not actual board size scale.



YARC ID Timer Project 20160404



YARC ID Timer Project 20160404

Software Downloads

Those wishing to modify the source code for the IDT, can find it for download at:

https://github.com/dtheriault/YARC.git

The software can be modified to work on a wide variety of Arduino based boards. Just modify the #define pin definitions to support your corresponding chip. In addition, changing the #define CALIBRATION value is probably required for timer accuracy.

Hardware files, schematic and PCB layout files are also included in the public repository for anyone wishing to generate your own board or modify the schematic accordingly. The hardware files were generated using Eagle v7.2.

All files are licensed under GPL and considered Open Source for use by anyone for any means. This means I am not responsible for any loss of equipment, eye strain or loss of hair follicles due to the assembly of SMT components or execution of this hardware or software.

Programming the ATtiny85

Programming of the ATtiny85 was simple using a USB Tiny AVR programming stick.

Tiny AVR Programmer from Sparkfun: PGM-11801

Club Members who wish to borrow one can contact me as I have (2) of them.

The Arduino environment is available for Windows, MAC OS X and Linux which I use. Downloads for your environment can be found from:
https://www.arduino.cc/en/Main/Software

The arduino.cc site has loads of information, tutorials and reference material to help you program using the Arduino environment.

Contact Info

Any questions/comments on the PCB Prototyping talk or the ID Timer including YARC members who need assistance with construction or programming of the project, feel free to email me via:

no1d.doug@gmail.com

73,
Doug - NO1D