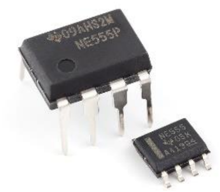
The 555 Timer Versus an ATTiny85, version 1.0

By R. G. Sparber

Protected by Creative Commons.[1]



The 555 Timer has been the work horse for generating timed voltage pulses since 1971[2]. Add a few resistors and capacitors to get pulse widths and pulse rates over a wide range. However, longer than a few seconds starts to sacrifice accuracy and above a minute demands special capacitors. Yet, this device remains the go-to answer to most analog timing problems. Depending on packaging and vendor, the cost can be around 40 cents in single lots. The added resistors and capacitors might add 30 cents.

Then we have the ATTiny. What an absurdity! You want to replace the lovely 555 with a computer system on a chip? Insanity, overkill, just plane offensive. On the other hand…

The cost of an ATTiny9-TS8R at Mousers is 34 cents in single lots. Although it has plenty of capability, I do not like dealing with its packaging.

The more capable ATTiny85 comes in at $1.24 for the through-hole[3] version. It is the same physical size as the 555 timer. I keep a stock of ATTiny85s in my shop.

Now, there is a a start-up cost since you must have a PC and buy a programmer. Sparkfun sells such a programmer for $15 plus shipping and handling (https://www.sparkfun.com/products/11801 . Download the free software development software, connect a USB cable, and you are ready to go.

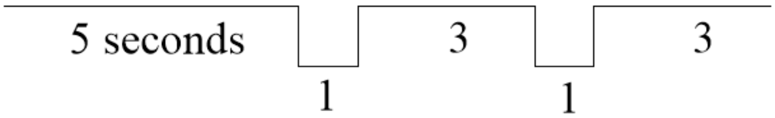
[2] https://en.wikipedia.org/wiki/555_timer_IC#History

[3] This means it is has pins that pass through the circuit board. The alternative is Surface Mount Technology (SMT) with tends to have much smaller soldering points but also lower cost.

No external parts are needed[4] and the ATTiny can run on any voltage between 2.7V and 5.5V. This means that 3 AA batteries is all you need.

Doing a fair comparison, the ATTiny can be a simple timer with a range of about 100 microsecond up to years. Ah, but it can do so much more even if I limit my discussion to just timing. Since it has 5 input/output pins, you can have 5 different timers running. Each timer can have a complex pulse pattern.

For example, consider making a single timer with a complex output pattern. When power is applied, we will start with an output near the battery voltage, delay for 5 seconds, change to a voltage near 0 volts for 1 second, then go back to near the battery voltage for 3 seconds. Repeat. Then repeat the entire sequence.



The 555 can be set up to output a give pulse rate and pulse width. Multiple 555s are needed to generate this complex pulse pattern.

Looking at the ATTiny's software, we can see what it takes to generate this signal. First we must tell the ATTiny85 that we want logical pin 0 to be our output and that it should start out near the battery voltage:

```
void setup(){ //this defines the start of the set up commands
pinMode(0, OUTPUT); //define logical pin 0 to be an OUTPUT
digitalWrite(0, HIGH); //set our output to be near the battery voltage
}
```

Logical pin 0 is physical pin 5 according to the ATTiny85's spec sheet[5].

[4] The Reset input does have an internal pull up resistor but some people like to add an external 10K to protect against noisy environments.

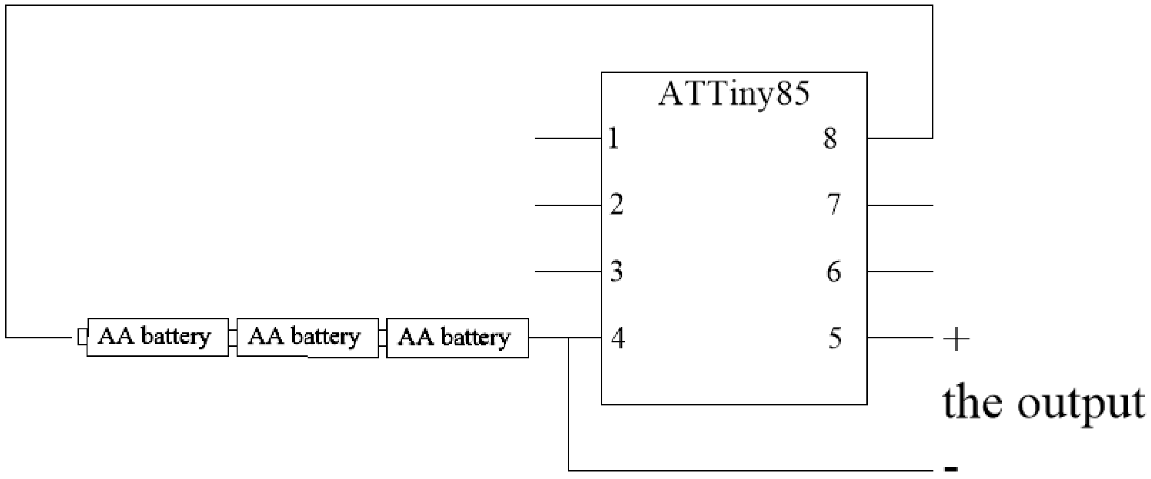
[5] https://www.mouser.com/datasheet/2/268/Atmel-2586-AVR-8-bit-Microcontroller-ATtiny25-ATti-1315542.pdf

Then we tell the ATTiny85 about the timing pattern. I have used the smallest variety of programming commands so as to not lose too many non-software types:

```
void loop(){ //this is the top of an endless loop

    delay(5000); //wait 5,000 milliseconds which is 5 seconds

        digitalWrite(0,LOW); //set the output to near 0 volts
        delay(1000); //wait 1000 milliseconds which is 1 second
        digitalWrite(0,HIGH); //then set the output to near the battery voltage
        delay (3000); // wait 3,000 milliseconds which is 3 seconds

        //then do the above pulse sequence a second time

        digitalWrite(0,LOW);
        delay(1000);
        digitalWrite(0,HIGH);
        delay (3000);

    } //this tells the ATTiny85 to go back to the top of the loop
```

The above program can be written with far fewer lines if the full set of commands is employed. But if all you want is to make a timer, this is all you need to know.

Once the program is loaded into the ATTiny85 using the programmer, we are ready to wire up our timer.



Three AA batteries supply the power to the ATTiny85. Its output is between pin 5 and pin 4. If desired, you can connect a 10K resistor between pin 1 and pin 8. This reduces the chance that stray electrical noise will bother the device.

Using the ATTiny85 as a single timer is sort of like pulling a Maserati with a plow horse. It will move down the road but, oh, the car can do so much more.

My hope is that you will give the ATTiny a try and become hooked. And if you need more input/output pins, there is a family of these devices in larger sizes. Both Sparkfun.com and Adafruit.com sell and support these devices. My favorite larger size is the Pro Micro which has the programmer built in. Sparkfun sells it for $20 and you only need a USB cable to program it.

I welcome your comments and questions.

If you wish to be contacted each time I publish an article, email me with just "Article Alias" in the subject line.

Rick Sparber
Rgsparber.ha@gmail.com
Rick.Sparber.org