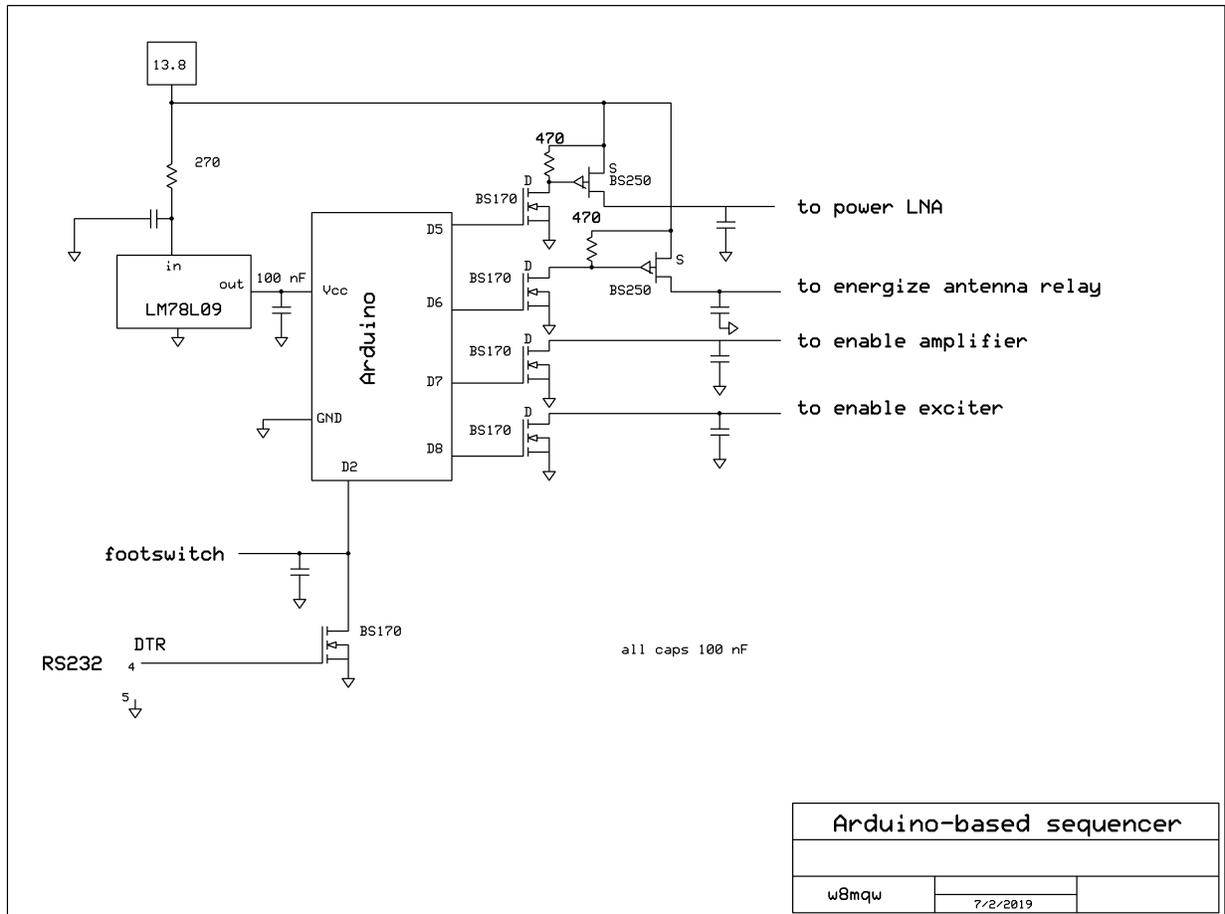


# An Arduino-based sequencer

Chuck MacCluer w8mqw

Microwave equipment requires a sequencer to carefully control the timing of the many independent modules cabled (cobbled?) together to form the whole [1],[2]. I present here what may be the simplest (and cheapest) possible four-level sequencer:



When a footswitch is pressed (or a COM port DTR goes high) the ports spool up --- the Arduino removes power from the LNA, energizes the changeover antenna relay to transmit, enables the power amplifier, and finally energizes the exciter. When the footswitch is released (or the COM port goes low), the sequencer spools down, reversing its previous steps. The intervals between each of the steps (in milliseconds) can be selected by merely changing the correspond line of code for that delay and then recompiling. The number of control ports can be easily expanded.

The Arduino code, which may be even simpler than the schematic, is displayed below.

[1] [http://www.w1ghz.org/seq/Why do I need a Sequencer.pdf](http://www.w1ghz.org/seq/Why%20do%20I%20need%20a%20Sequencer.pdf)

[2] [http://www.w1ghz.org/seq/Even More Fool-resistant Sequencer Mk3.pdf](http://www.w1ghz.org/seq/Even%20More%20Fool-resistant%20Sequencer%20Mk3.pdf)

## The script

```
/******  
const int footswitch = 2; // the number of the footswitch/COM pin  
const int LNA = 5;  
const int AntRelay = 6;  
const int PwrAmp = 7;  
const int PTT = 8;  
int state = 1;  
int prevstate = 1;  
/******  
void setup() { // initialize the pin functions:  
  pinMode(footswitch, INPUT_PULLUP);  
  pinMode(LNA, OUTPUT);  
  pinMode(AntRelay, OUTPUT);  
  pinMode(PwrAmp, OUTPUT);  
  pinMode(PTT, OUTPUT);  
  //intialize their values:  
  digitalWrite(footswitch,HIGH);  
  digitalWrite(PTT,LOW);  
  digitalWrite(PwrAmp,LOW);  
  digitalWrite(AntRelay,LOW);  
  digitalWrite(LNA,HIGH);    //turn on the LNA  
} /******
```

```

void loop() { //We poll to see if a change of state has occurred
  state = digitalRead(footswitch);
  if (state == prevstate)delay(25); else
    { if (state == 0) spoolup();
      if (state == 1) spooldown();
      prevstate = state; // thus nothing will happens until a future change of state
    }
}

/*****FUNCTIONS*****/

void spoolup() {
  digitalWrite(LNA, LOW); //power to LNA is removed
  delay(25); //can adjust each delay (in milliseconds) to match system timing
  digitalWrite(AntRelay,HIGH); //antenna relay is energized
  delay(25);
  digitalWrite(PwrAmp,HIGH); //power amp is enabled
  delay(25);
  digitalWrite(PTT,HIGH); //exciter is enabled
  delay(25);
}

/*****/

void spooldown() {
  digitalWrite(PTT,LOW); // exciter is disabled
  delay(25);
  digitalWrite(PwrAmp,LOW); //power amp is disabled
  delay(25);
  digitalWrite(AntRelay,LOW); //antenna relay is opened
  delay(25);
  digitalWrite(LNA,HIGH); // LNA is repowered
  delay(25);
}

```