# Tomahawk Missiles
# IR Beacons
# State Machines
# and
# Why I Hate Angles

# Tomahawk Missile

- Laser guided
- Someone puts a pulsed laser beam on target.
- Someone releases missile
- Missile hunts for pulsed laser
- Once found, it homes in on target

# Tomahawk Missile

- Laser pulses are very short in duration
  - Handfull of nanoseconds
- Pulses happen in the 10 Hertz rate
- Duration between pulses can be:
  - Fixed – PRF (256 codes)
  - Variable – PIM (coded for extra security)
- Timing of pulses is **very** accurate
- The exact PRF or PIM values change every mission

# Tomahawk

- The receiver knows the expected duration between pulses.
- When it sees a pulse it computes the time the next pulse should arrive.
- It computes a small window of time
- If a pulse arrives in that window, it increments a counter
- If a pulse does not arrive, it resets the counter.
- If the counter gets above N, the seeker is locked.

# IR Beacon

- 4 IR LEDs arranged for full 360 horizontal coverage.
- Not a single pulse like the laser, but rather 500 microseconds of pulsed light.
- Bursts of pulsed light happen in the 10 Hz range.
- DIP switch allows you to select one of 8 codes.
- Currently a PIC is running the show.
    - Have plans to migrate to an AtTiny85

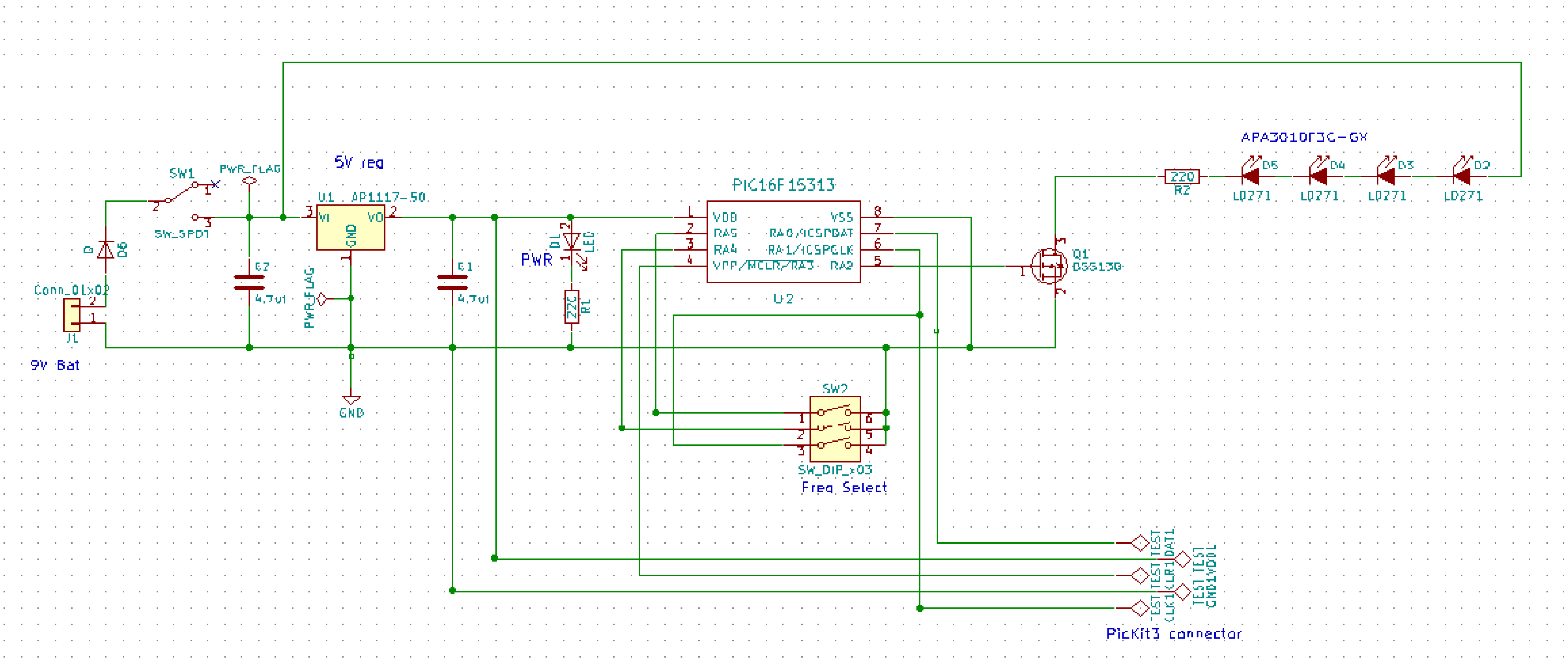
# Receiver

- Standard 38kHz receiver
- Receiver is in a tube to limit feild of view
  - Black PLA is transparent to IR.
- 5V, ground, and a digital signal.
  - Signal is high in presense of 38kHz IR
- Signal runs to an interrupt pin on Arduino

# Receiver

- ISR looks to see if pulse is in window
- ISR manages the window timer and counter
- ISR calls a routine when lock status changes
- One routine in 'loop' needs to be called to clear counter should signal disappear completely.
- User can set the desired code to seek
- User can call routine to get lock status

# IR Beacon Schematic

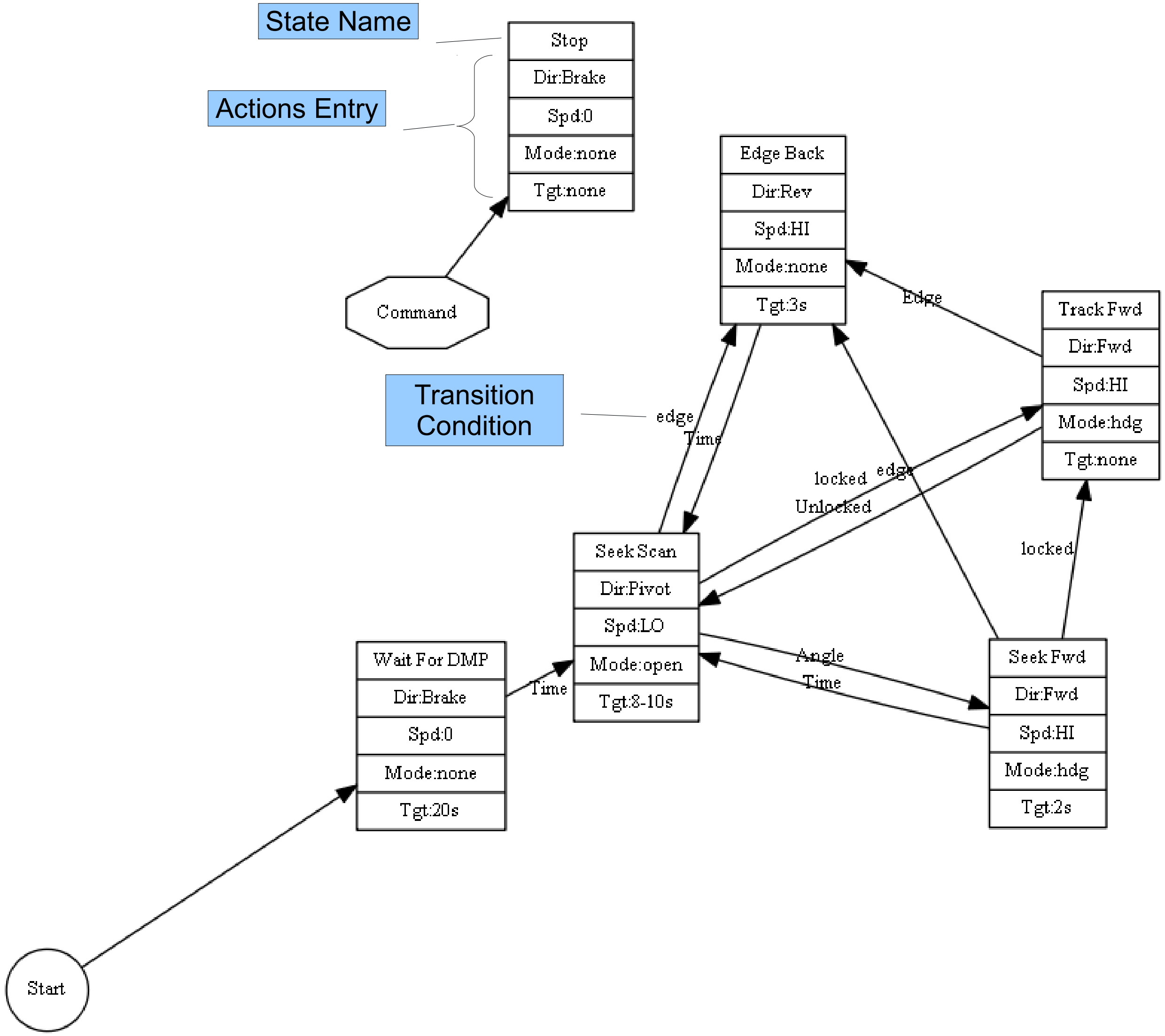


# Issues

- Try to lower cost to bellow $10
- Migrate to AtTiny85
  - On PIC, all code is done by HW or ISR.
- Add reverse protection diode
- Add standard AVR pad layout for pogo pin programmer

# DEMO

# State Machine



Rover State Machine

# Auto Generated

- Using Graphviz
- Add Graphviz/Neato code as comments to code
- Run a simple Python program to strip out comments
- Run 'Comments' through Neato to generate graph
- Able to document in the code.
- Easy to update
- Great for finding logic bugs in complex FSMs

# Neato Header

```
//FSM: digraph FSM {
//FSM: rankdir=LR;
//FSM: fontsize = 10;
//FSM: size="8,10";
//FSM: page="8.5,11";
//FSM: start=1;
//FSM: overlap=scale;
//FSM: splines=true;
//FSM: orientation=portrait;
//FSM: sep=0.5;

//FSM: Start   [label="Start",                  shape=circle,  fontsize=10];
//FSM: Command [label="Command",    shape=octagon, fontsize=10];
//FSM: Stop         [label="Stop|Dir:Brake|Spd:0|Mode:none|Tgt:none",      shape=record,  fontsize=10];
//FSM: S_WFD   [label="Wait For DMP | Dir:Brake|Spd:0|Mode:none|Tgt:20s", shape=record,  fontsize=10];
//FSM: S_SF    [label="Seek Fwd|Dir:Fwd|Spd:HI|Mode:hdg|Tgt:2s",                shape=record,  fontsize=10];
//FSM: S_SS    [label="Seek Scan|Dir:Pivot|Spd:LO|Mode:open|Tgt:8-10s",      shape=record,  fontsize=10];
//FSM: S_TF    [label="Track Fwd|Dir:Fwd|Spd:HI|Mode:hdg|Tgt:none",           shape=record,  fontsize=10];
//FSM: S_EB    [label="Edge Back|Dir:Rev|Spd:HI|Mode:none|Tgt:3s",            shape=record,  fontsize=10];
//FSM: Start -> S_WFD;
//FSM: Command -> Stop;
```

# Neato State Transition

```
switch (State) {

  // Robot is waiting for DMP to stabilize
   case WAIT_FOR_DMP:

      // When time expires
      if (micros() > targetTime)
      {
         State = SEEK_SCAN;
         //FSM: S_WFD -> S_SS [label="Time",fontsize=10];
      }
      break;

      ...
}
```

# Neato Footer

```
//FSM: fontsize  = 14;
//FSM: label     = "Rover State Machine";
//FSM: }
```

# Generating Graph

- Small Python script pulls out all lines with //FSM:
- It deletes the prefix and appends result to file
- The resulting file is passed to Neato to generate the graph. (Pdf, Png, ...)
- See MakePlot.py in source

# Why I Hate Angles

- Radians, degrees, artilery mils, ...?
- +- 180 or 0 to 360?
- Maybe only +-90 for latitudes?
- Issues with adding or subtracting
    - 340 + 30 = 370  -> 10

# Normalizing

```
float normalize(float ang)
{
    if (ang > 360) ang = ang – 360;
    if (ang < 0) ang = ang + 360;
    return ang;
}
```

# Use

```
Hdg = 340;
Bias = 340;
Correction = 300;
Hdg = hdg + bias + correction;
Hdg = normalize(hdg);
```

Print HDG gives 620 not 260.

# Normalize (Bad Version)

```
float normalize(float ang) {
  while (true) {
    if (ang > 360) {
      ang = ang – 360
    } else if (ang < 0) {
      ang = ang + 360
    } else {
      break;
    }
  }
  return ang;
}
```

Dangerous!
What happens if ang is huge?
Ask me how I know!

# Normalize Safe Version

```
float normalize(float ang, int n) {
  for (i=0; i<n; i++) {
    if (ang > 360) {
      ang = ang – 360
    } else if (ang < 0) {
      ang = ang + 360
    } else {
      break;
    }
  }
  return ang;
}
```

Might want to add some Asserts here to catch when you still have a bad angle.

# Use

```
Hdg = 340;
Bias = 340;
Correction = 300;
Hdg = hdg + bias + correction;
Hdg = normalize(hdg, 2);
```

Print HDG gives 260.

# Some Angles are not Normal!

- Driving up a parking lot corkscrew ramp, you may turn more than 360 degrees!
- To got back down you need to know how many turns to make.
- Your MPU-6050 compass only returns 0-360.
- This is a tough nut I am still working on.
- How to use a normal compass heading to turn 360 or more degrees.
- Suggestions welcome.

# Questions?

- Code will be posted to my website shortly.
- Can find link at our wiki
    - http://www.nashuarobotbuilders.org/wikid/pmwiki/pmwiki.php
- At my website:
    - Www.fll-freak.com