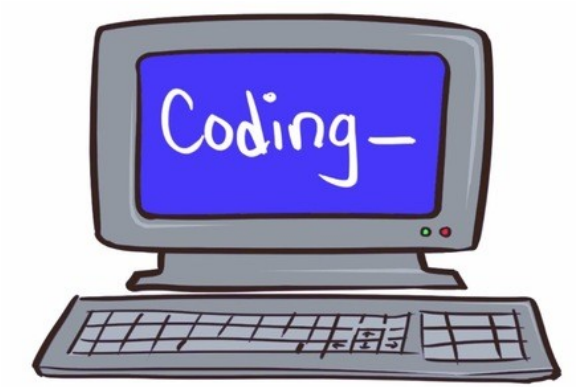


Class 45

“STATE MACHINES”

applied to

HC-SR04 sonar ranger

sketches

Sonar1OnDemand.ino

- 1-shot. Each “ping” called as needed.

Sonar1Contin.ino

- Nonstop pinging.

Sonar1Flag.ino

- Continuous. Issue an alert flag if “too close”.

SonarMulti.ino

- Coordinated modules all running continuous.

The arduino model – the “delay” problem

```
#include "lib2.h"


void setup() {
    Serial.begin(9600);
}

void loop() {
    readKeyboard();
    quadencoder1.run();
    Serial.print("boo");
    delay(1000);
}
```

Very fast “.run()” calls are one attempt to avoid delays in any area of code.

Any delay() in loop() or in any called function, will suspend all other processing.

Ztimer

A way to get a delay without "blocking"

```
#include "ZTimer.h"
ZTimer timr2;

void setup() {
  pinMode(LED_BUILTIN, OUTPUT);
  Serial.begin(115200);
  Serial.println("Testing ZHomeslices ZTimer Library Basic Simple");
  timr2.SetWaitTime(1000); //Timer will trigger at 1 second intervals
  timr2.ResetTimer(true);  // reset the timer with repeat flag true and enable
}

void loop() {
  if (timr2.CheckTime() == true) { // Check to see if time has expired
    Serial.println(" The Basic Way");
    digitalWrite(LED_BUILTIN, !digitalRead(LED_BUILTIN));  // Blink onboard LED
  }
}
```

.run()

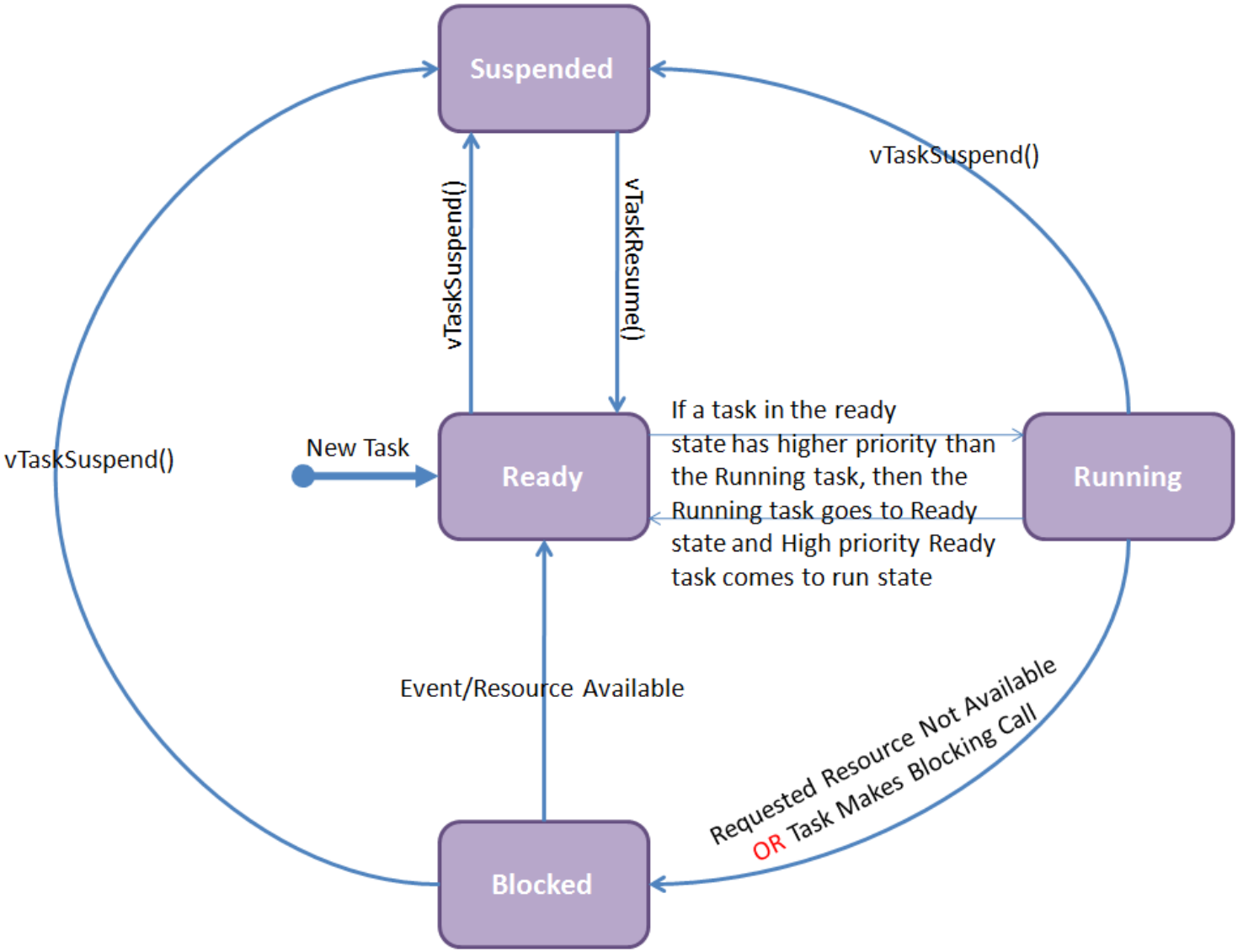
VERY short call made each time through loop()

Examples of “.run()”

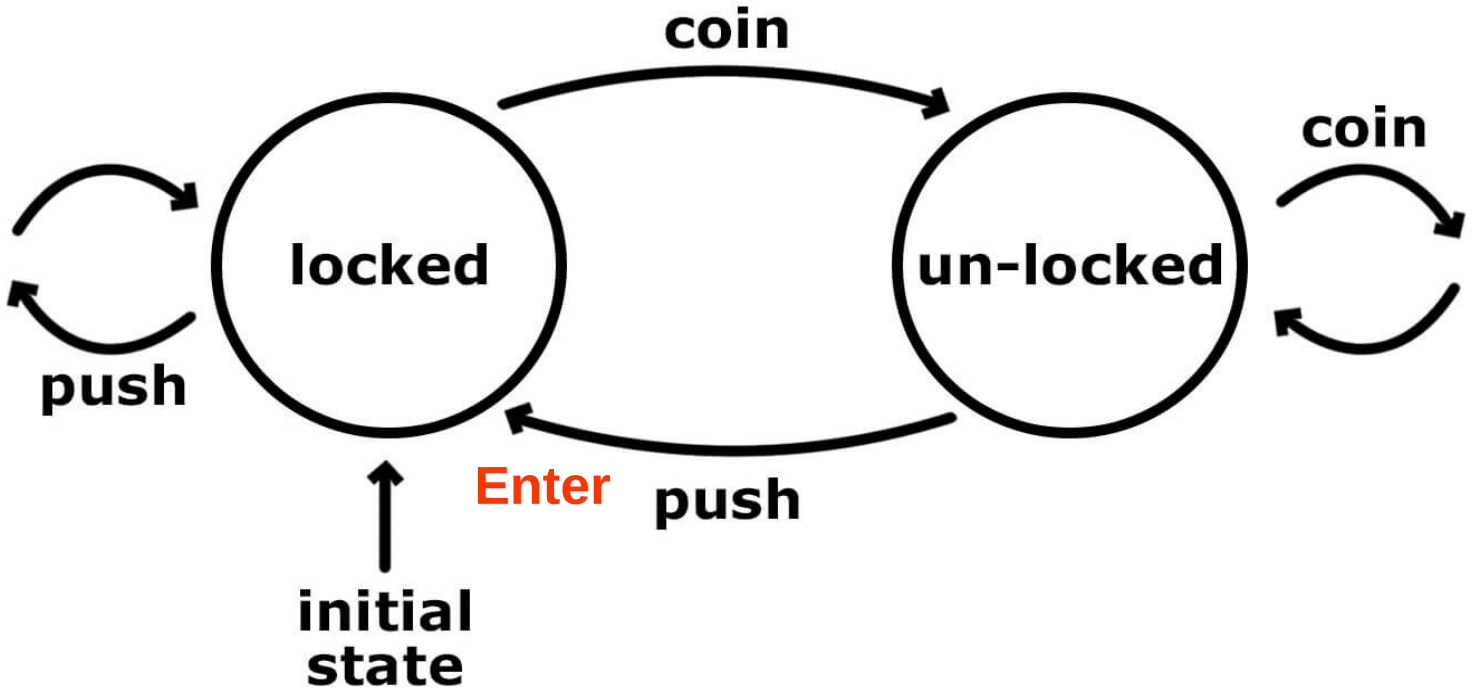
- Blynk.run()
- wifiWatch()
- QuadEncoder.run()
- Sonar.run() (today)

.run() usually needs a “state machine” coding.

Any task may be running or paused



Turnstyle
... a state machine



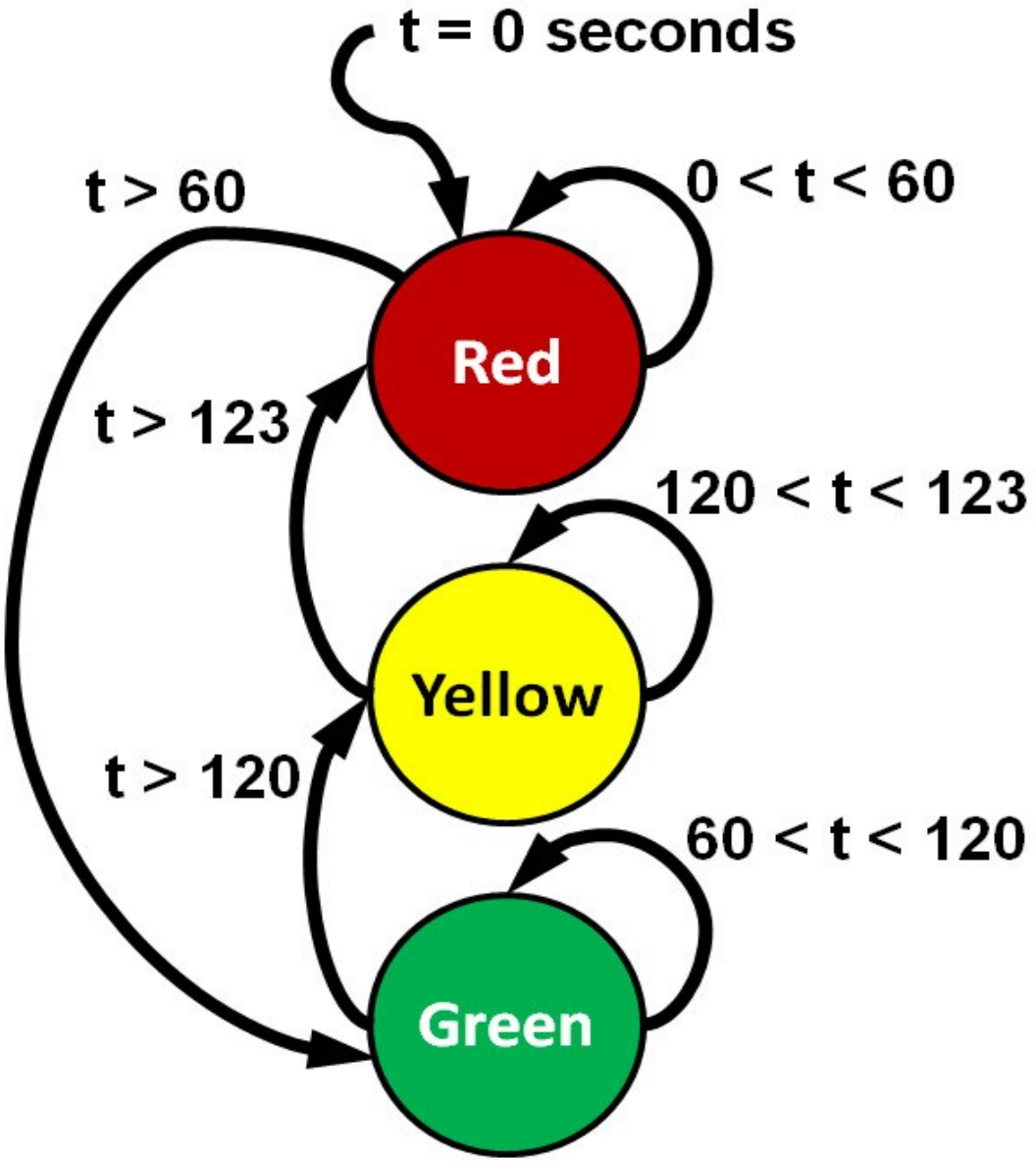
State machines

https://www.youtube.com/watch?v=hJIST1cEf6A

https://www.youtube.com/watch?v=-Yicg2TTMPs

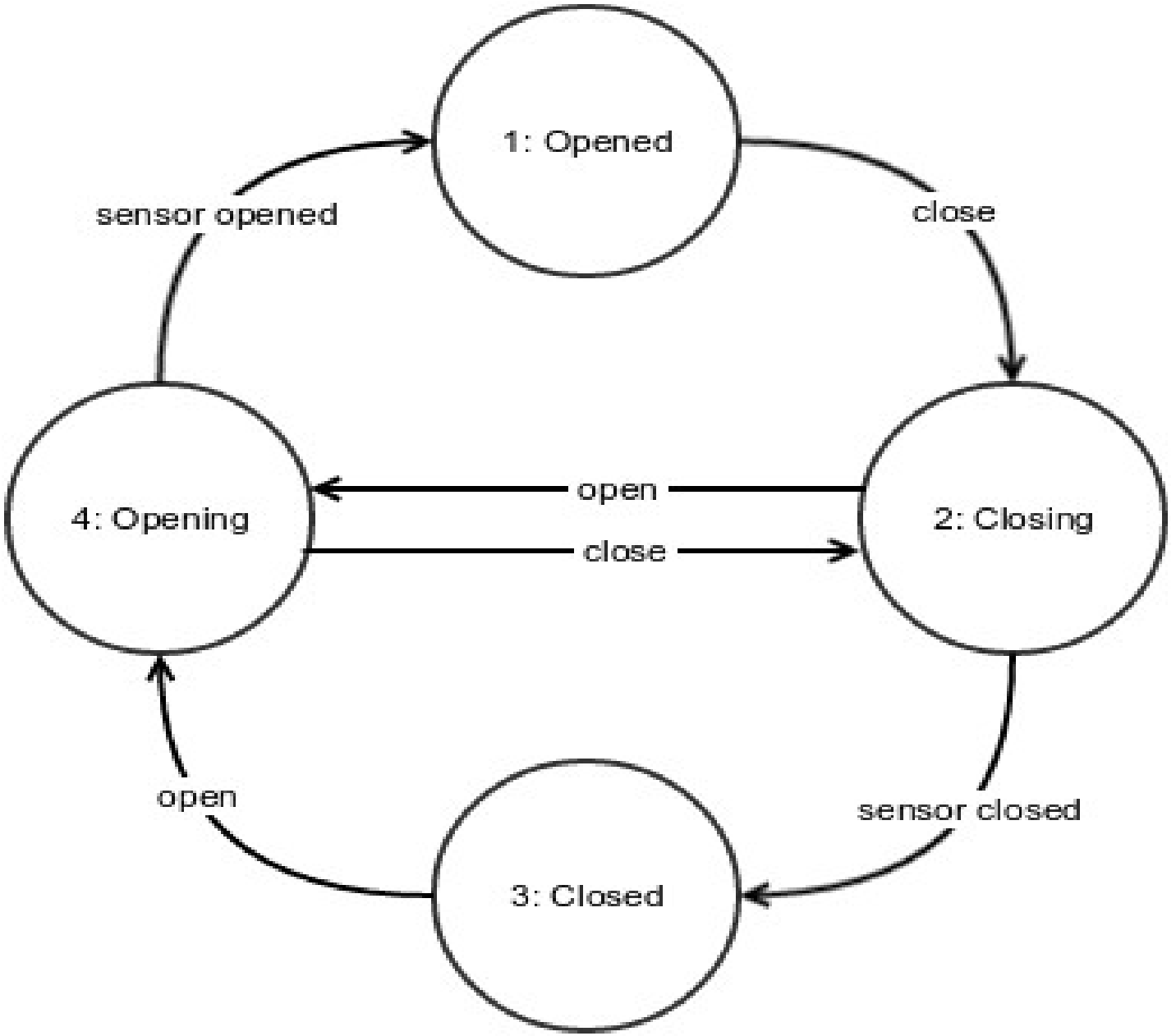
https://www.youtube.com/watch?v=0z0XGXBtI3A

Traffic Light



Traffic Light

In this model, the “event” or “trigger” causing transitions is simply a clock counting up to 123 seconds.



Property Gate

Takes 20 secs to operate.

Inputs:

- 1 control switch saying OPEN/CLOSE
- 1 limit switch for gate fully closed
- 1 limit switch for gate fully open

In general it is possible to encode into a lookup table all “events” vs all “states” to determine the new state

State transition table

Input \ Current state	State A	State B	State C
Input X	...	...	...
Input Y	...	State C	...
Input Z	...	...	...

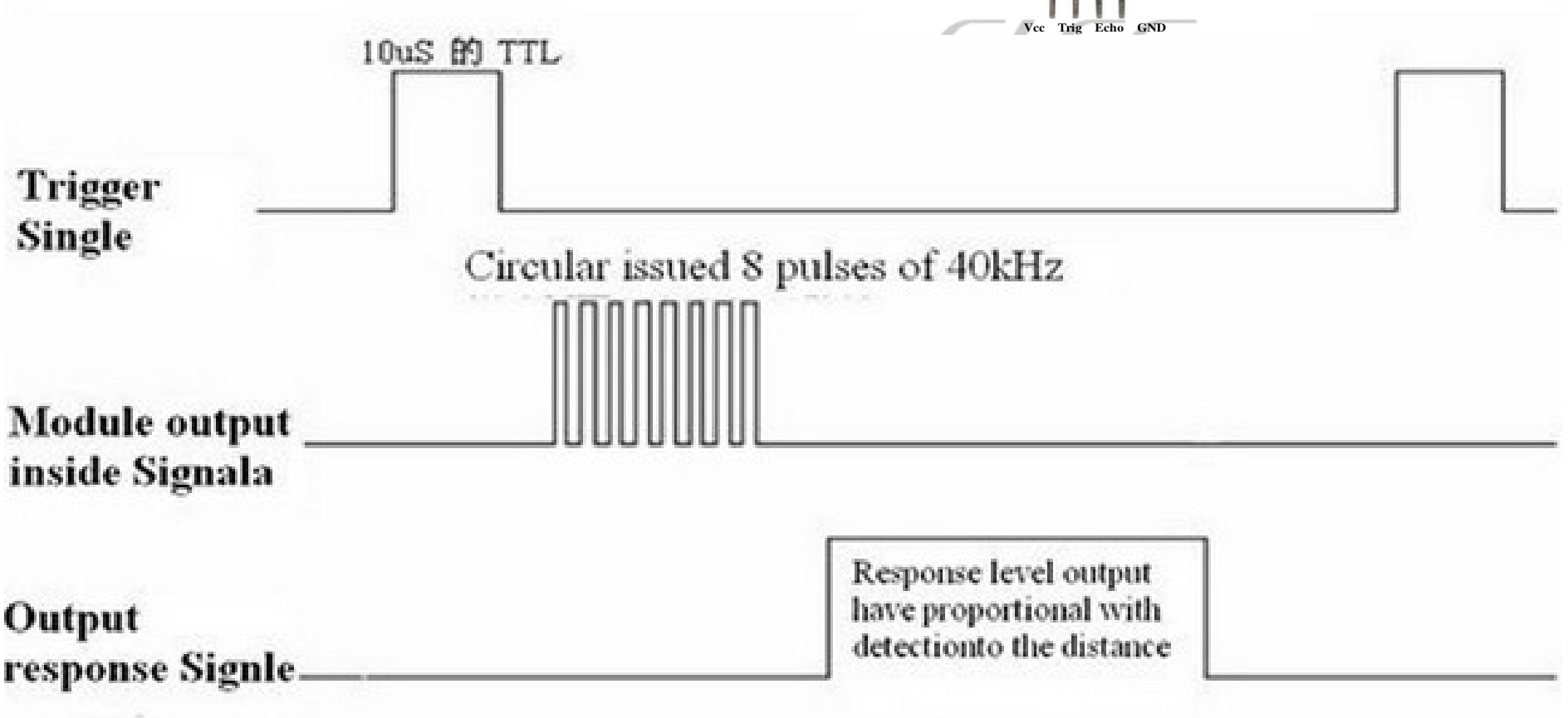
Some state table programs do use this sort of lookup table.

The quad encoder lookup table was a version of state table

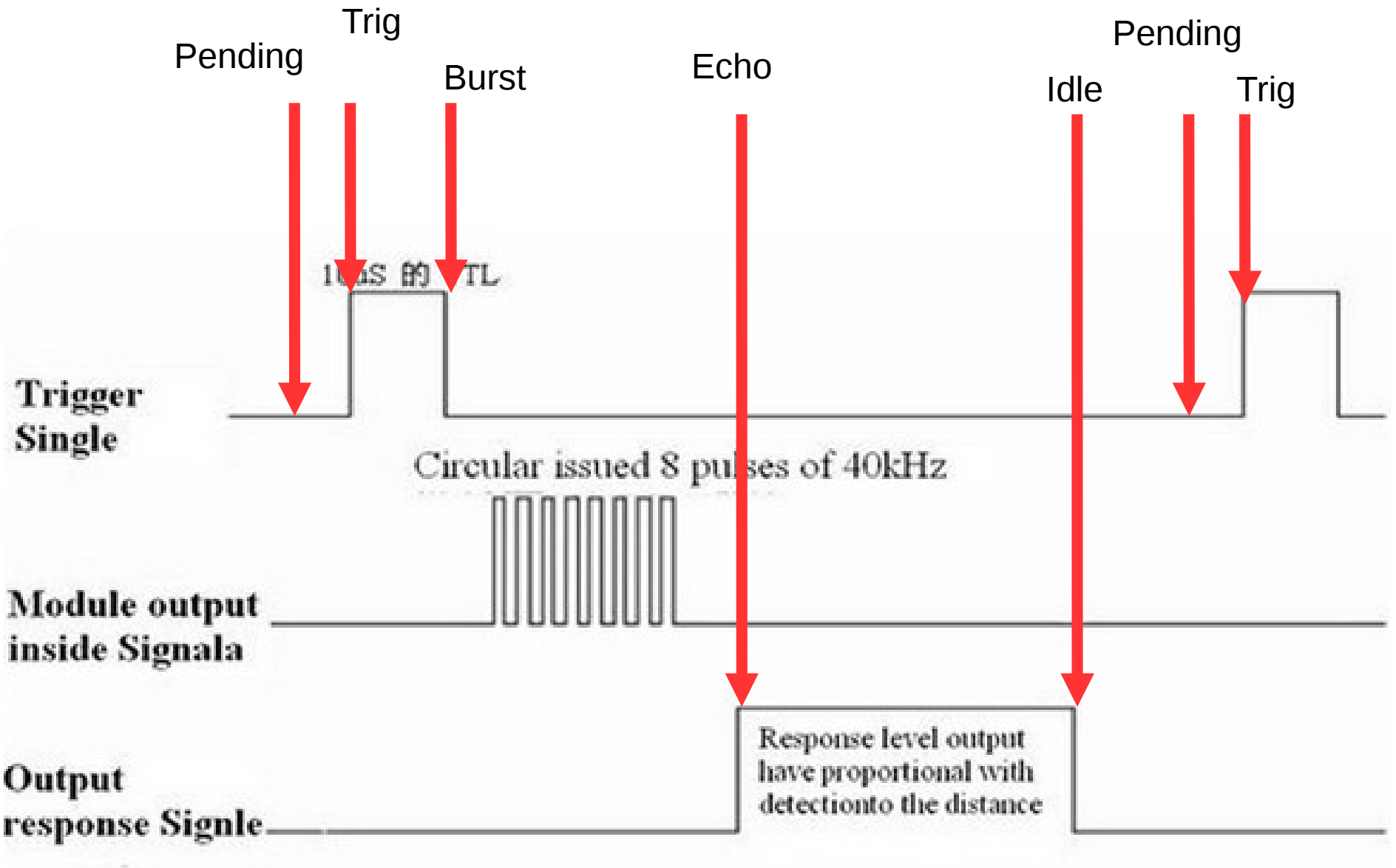
```
int QuadEncoder::_encoderStates[4][4] = {
        {  0, -1,  1,  0 } ,
        {  1,  0,  0, -1 } ,
        { -1,  0,  0,  1 } ,
        {  0,  1, -1,  0 }    };


QuadEncoder::QuadEncoder(void)
{
    _state=0;
    _prevState=0;
    _debounceMicros = 0L;
    _counter = 0L;
    _active = false;
    debounce = 0;
}
```

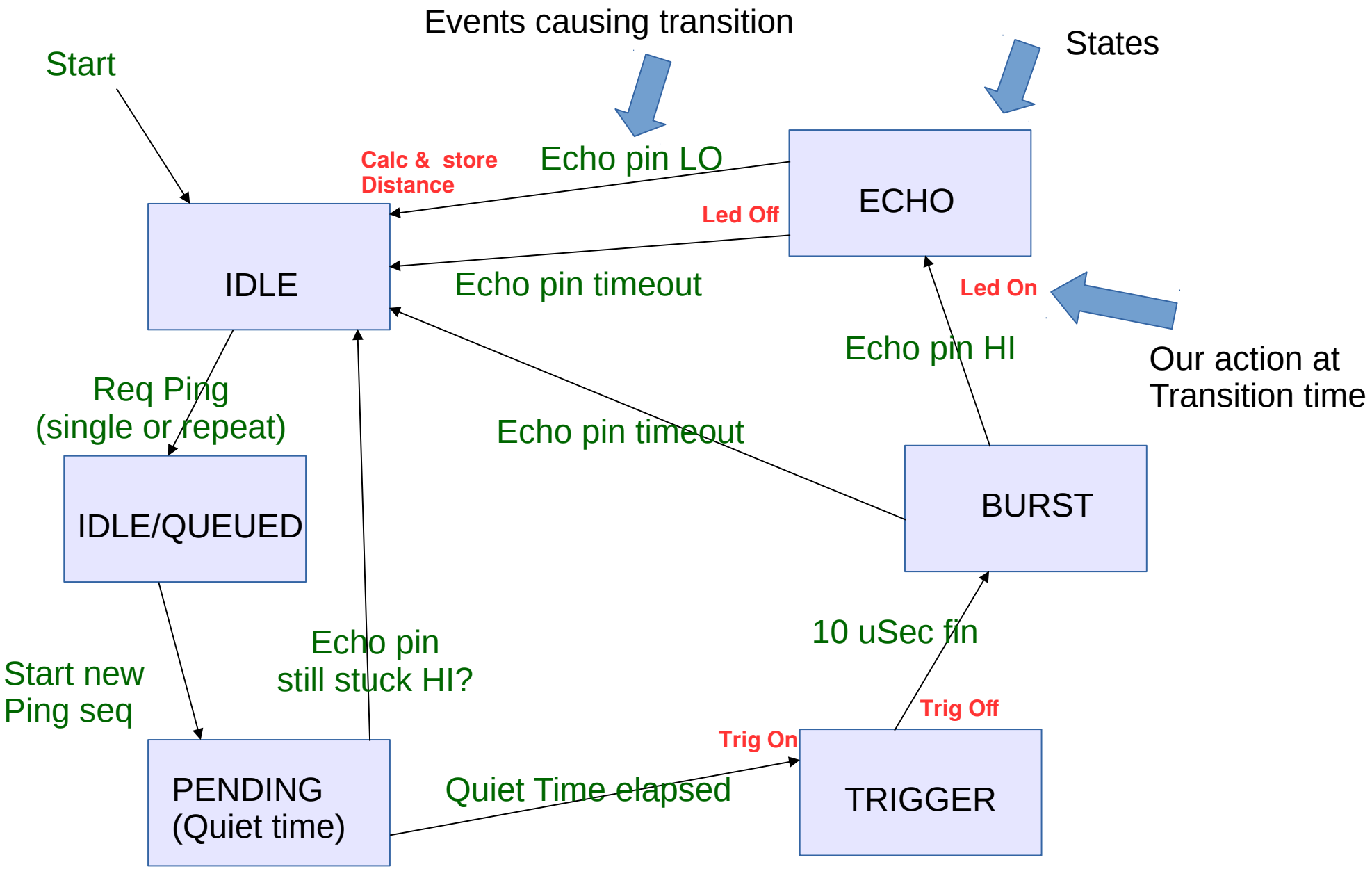
Actually, this state table was computing not the next state,
but rather an “output”
to be added to counter.

Timing Diagram HC-SR04



STATES


Pending
Trig
Burst
Echo
Idle
Pending
Trig
Trigger
Single
Circular issued 8 pulses of 40kHz
Module output
inside Signala
Output
response Signle
Response level output
have proportional with
detectionto the distance


Events causing transition
States
Start
Calc & store Distance
Echo pin LO
ECHO
Led Off
IDLE
Echo pin timeout
Led On
Echo pin HI
Our action at Transition time
Req Ping (single or repeat)
Echo pin timeout
IDLE/QUEUED
BURST
10 uSec fin
Start new Ping seq
Echo pin still stuck HI?
Trig Off
Trig On
PENDING (Quiet time)
Quiet Time elapsed
TRIGGER

```
case state_burst:
    if (digitalRead(_echoPin) == HIGH)
    {
        // echo mode pulse has started
        // empirically, seems to be about 450 uSec af
        if (_led)
            digitalWrite(abs(_led), (_led<0));
        _state = state_echo;
        _maxTime = t + _maxEchoTime;
        _echoStartTime = t;
        return true;
    }
    if((t - _maxTime) >= 0L)  // timeout
    {
        // Failed to start echo mode. Simply give up
        _state = state_idle;
        isPinging = false;
        return false;
    }
    return true;   // still waiting! Yield until nex

case state_echo:
    if(digitalRead(_echoPin) == LOW)
    {
        // successful end to echo pulse
        _distance = ((int)(t -  _echoStartTime)) / US
        if(_led)
            digitalWrite(abs(_led),(_led>0));
```

Current state

Transition trigger or “event”

New state

Transition trigger or “event”

New state

HC-sr04