September 19,2024 The state of the project And thanks for reading this.

The detector could be described as a ‘quantum detector’ It detects the movement of magnetic north in quantum increments about two minutes apart (where there are 60 minutes in a degree. They are about two millimeters apart.)

A realistic upgrade would be to evaluate magnetic north from a careful evaluation of the behavior of a gently swinging needle. For Example: the magnetic needle is swinging back and forth (should be simple harmonic motion) perhaps with an amplitude of 40 centimeters. The path would be carefully lined with LDRs. The LDRs would be linked to a physical location. This is much the same as the present system. ( I have named the present platform Shelly) In the new platform (Racheal) the lDRs would be spread across the 40 centimeters and the LDRs would be carefully assigned a location. So, Shelly and Racheal would be very very similar regarding the physical apparatus.

The Racheal software would average the stimulated LDRs and that would be the measurement of magnetic north. This could be exceedingly complex regarding the software, but maybe just an average would work fine. Racheal needs to be made and the software written and then the two platforms can be run parallel. There could be infinite fuss and aggravation, but maybe this , in its simplest form, would work?

Example: the needle is perturbed in some way. (this is all done in the program and the magnitude is easy to control)

The needle swings and every LDR is counted and the total summed. After an interval set by the program and easily changed, an average is calculated. This side could be either in the C++ or in the Python. My bias is to go with the C++. Five minutes later it would repeat.

I ordered another set of magnets. I built another needle apparatus. I made a video as I was making it then just threw together a short video. https://youtu.be/Tpm7I1Sw9l4

Shelly, the running operational system, has 33 ‘if’ statements to find the stimulated LDR. I feel like this aspect of the programming should be cleaned up. I went to my C++ books and came up with this:

#define ldrPin53 53 // These must be on different lines ,,,, okay

#define ldrPin52 52

#define ldrPin51 51

#define ldrPin50 50

#define ldrPin49 49

#define ldrPin48 48

#define ldrPin47 47

#define ldrPin46 46

#define ldrPin45 45

#define ldrPin44 44

#define ldrPin43 43

#define ldrPin42 42

#define ldrPin41 41

#define ldrPin40 40

#define ldrPin39 39

#define ldrPin38 38

#define ldrPin37 37

#define ldrPin36 36

#define ldrPin35 35

#define ldrPin34 34

#define ldrPin33 33

#define ldrPin32 32

#define ldrPin31 31

#define ldrPin30 30

#define ldrPin29 29

#define ldrPin28 28

#define ldrPin27 27

#define ldrPin26 26

#define ldrPin25 25

#define ldrPin24 24

#define ldrPin23 23

#define ldrPin22 22

#define relayPin 8

#define puffPin 9

boolean ldrstatus53 = 0; boolean ldrstatus52 = 0;boolean ldrstatus51 = 0;boolean ldrstatus50 = 0;boolean ldrstatus49 = 0;

boolean ldrstatus48 = 0;boolean ldrstatus47 = 0; boolean ldrstatus46 = 0; boolean ldrstatus45 = 0; boolean ldrstatus44 = 0;

boolean ldrstatus43 = 0; boolean ldrstatus42 = 0;boolean ldrstatus41 = 0;boolean ldrstatus40 = 0;boolean ldrstatus39 = 0;

boolean ldrstatus38 = 0; boolean ldrstatus37 = 0;boolean ldrstatus36 = 0;boolean ldrstatus35 = 0;boolean ldrstatus34 = 0;

boolean ldrstatus33 = 0; boolean ldrstatus32 = 0;boolean ldrstatus31 = 0;boolean ldrstatus30 = 0;boolean ldrstatus29 = 0;

boolean ldrstatus28 = 0; boolean ldrstatus27 = 0;boolean ldrstatus26 = 0;boolean ldrstatus25 = 0;boolean ldrstatus24 = 0;

boolean ldrstatus23 = 0; boolean ldrstatus22 = 0;

float loc\_53 = 181.12; float loc\_52 = 179.04;float loc\_51 = 176.96;float loc\_50 = 174.88;float loc\_49 = 172.8;float loc\_48 = 170.72;

float loc\_47 = 168.64; float loc\_46 = 166.56;float loc\_45 = 164.48;float loc\_44 = 162.4; float loc\_43 = 160.32;float loc\_42 = 158.24;

float loc\_41 = 156.16; float loc\_40 = 154.04;float loc\_39 = 152.00;float loc\_38 = 149.92;float loc\_37 = 147.84;float loc\_36 = 145.76;

float loc\_35 = 143.68; float loc\_34 = 141.6; float loc\_33 = 139.52;float loc\_32 = 137.44;float loc\_31 = 135.36;float loc\_30 = 133.28;

float loc\_29 = 131.2; float loc\_28 = 129.12;float loc\_27 = 127.04;float loc\_26 = 124.96;float loc\_25 = 122.88;float loc\_24 = 120.8;

float loc\_23 = 118.72; float loc\_22 = 116.64;

boolean pin\_status[40] = {

ldrPin53, ldrPin52,ldrPin51,ldrPin50,ldrPin49,ldrPin48,ldrPin47,

ldrPin46,ldrPin45,ldrPin44,ldrPin43,ldrPin42,ldrPin41,ldrPin40,ldrPin39,ldrPin38,

ldrPin37,ldrPin36,ldrPin35,ldrPin34,ldrPin33,ldrPin32,ldrPin31,ldrPin30, ldrPin29,

ldrPin28,ldrPin27,ldrPin26,ldrPin25,ldrPin24,ldrPin23,ldrPin22 }

;

//// okay good, I need an array of values

float location\_values [40] = {

181.12, 179.04, 176.96, 174.88, 172.8, 170.72,

168.64, 166.56, 164.48, 162.4, 160.32, 158.24,

156.16, 154.04, 152.00, 149.92, 147.84, 145.76,

143.68, 141.6, 139.52, 137.44, 135.36, 133.28,

131.2, 129.12, 127.04, 124.96, 122.88, 120.8,

118.72, 116.64};

boolean bool\_hold=0;

void setup() {

pinMode (ldrPin53, INPUT); pinMode (ldrPin52, INPUT); pinMode (ldrPin51, INPUT); pinMode (ldrPin50, INPUT);

pinMode (ldrPin49, INPUT); pinMode (ldrPin48, INPUT); pinMode (ldrPin47, INPUT); pinMode (ldrPin46, INPUT);

pinMode (ldrPin45, INPUT); pinMode (ldrPin44, INPUT); pinMode (ldrPin43, INPUT); pinMode (ldrPin42, INPUT);

pinMode (ldrPin41, INPUT); pinMode (ldrPin40, INPUT); pinMode (ldrPin39, INPUT); pinMode (ldrPin38, INPUT);

pinMode (ldrPin37, INPUT); pinMode (ldrPin36, INPUT); pinMode (ldrPin35, INPUT); pinMode (ldrPin34, INPUT);

pinMode (ldrPin33, INPUT); pinMode (ldrPin32, INPUT); pinMode (ldrPin31, INPUT); pinMode (ldrPin30, INPUT);

pinMode (ldrPin29, INPUT); pinMode (ldrPin28, INPUT); pinMode (ldrPin27, INPUT); pinMode (ldrPin26, INPUT);

pinMode (ldrPin25, INPUT); pinMode (ldrPin24, INPUT); pinMode (ldrPin23, INPUT); pinMode (ldrPin22, INPUT);

Serial.begin(9600);

Serial.println("hello world");

}

void loop() {

for ( int counter= 0; counter <30 ; counter++ ){

bool\_hold= digitalRead(pin\_status[counter]);

if (bool\_hold==1){

Serial.println (counter);

Serial.println (bool\_hold);

Serial.println (location\_values [counter]);

Serial.println();

delay (300);

}

}

}

I have two Mega 2650 boards and an extra laptop. I wired up the board identically to Shelly, but left out the LDRs. So positive input is just touching the lead with a 5 V probe. I loaded all the software on the Notepad (I have given it a name) and the above program works. Just to the Arduino Serial monitor, the program prints the location of the stimulated probe. So the ‘For’ loop with the arrays is the sophistication I need in Shelly. For starters, I tried to put this addition into Shelly and could not make it work. (I was not terribly persistent.)

Then it occurred to me that my new program, the one above, could be loaded into the shelly platform. If nothing else it would be a great QC platform. It refused to work and the problem was software (it found random Boolean yes for the i/o pins) (again I was not terribly persistent)

I feel like there are spirits following me and telling me to chill, so that is what I am going to do (for now)

Shelly, of course, plugs along taking a reading every five minutes. The laser is driven by three AAA batteries. I change them out every three weeks. I am running the data into just one table in one database. Thee are presently about 45K rows of data.