ANEXO B – Algoritmo de Kelly

```
#define LENGTH 512

byte rawData[LENGTH];
int count;

// Sample Frequency in kHz
const float sample_freq = 8919;

int len = sizeof(rawData);
int i,k;
long sum, sum_old;
int thresh = 0;
float freq_per = 0;
byte pd_state = 0;

void setup(){
  analogReference(EXTERNAL);   // Connect to 3.3V
  analogRead(A0);
  Serial.begin(115200);
  count = 0;
}

void loop(){
  if (count < LENGTH) {
    count++;
    rawData[count] = analogRead(A0)>>2;
  } else {
    sum = 0;
    pd_state = 0;
    int period = 0;
    for(i=0; i < len; i++) {
      // Autocorrelation
      sum_old = sum;
      sum = 0;
      for(k=0; k < len-i; k++) sum += (rawData[k]-128)*(rawData[k+i]-
128)/256;
      // Serial.println(sum);

      // Peak Detect State Machine
      if (pd_state == 2 && (sum-sum_old) <=0) {
        period = i;
        pd_state = 3;
      }
      if (pd_state == 1 && (sum > thresh) && (sum-sum_old) > 0) pd_state =
2;
      if (!i) {
        thresh = sum * 0.5;
        pd_state = 1;
      }
    }
    // for(i=0; i < len; i++) Serial.println(rawData[i]);
```

```
    // Frequency identified in Hz
    if (thresh >100) {
      freq_per = sample_freq/period;
      Serial.println(freq_per);
    }
    count = 0;
  }
}
```