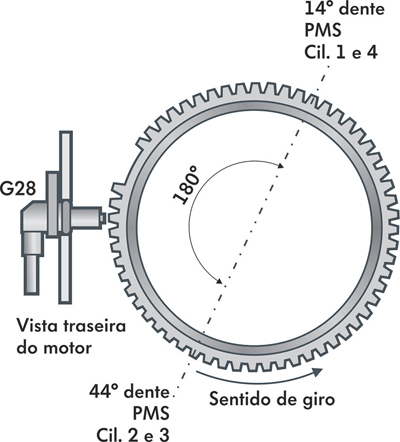
**Simulador de Sinal de Roda Fônica**

Com esse simulador pretende-se desenvolver na Plataforma Arduino Uno um programa que possa simular os sinais de rotação e fase em um motor de combustão do Ciclo Otto de 4 cilindros. O sinal será equivalente ao sinal vindo de um sensor do tipo Hall.

**Sinal de Rotação**

O sinal de rotação nesse tipo de motor é proveniente da leitura dos dentes de uma roda fônica solidária ao virabrequim do motor. A roda possui 60 dentes com os dois últimos dentes faltantes (Dente 59 e 60), para a identificação do PMS.



<http://www.noticiasdaoficinavw.com.br/v2/wp-content/uploads/2013/06/img-12.jpg> - Acesso em fev/2018

**Programação do sinal de Rotação**

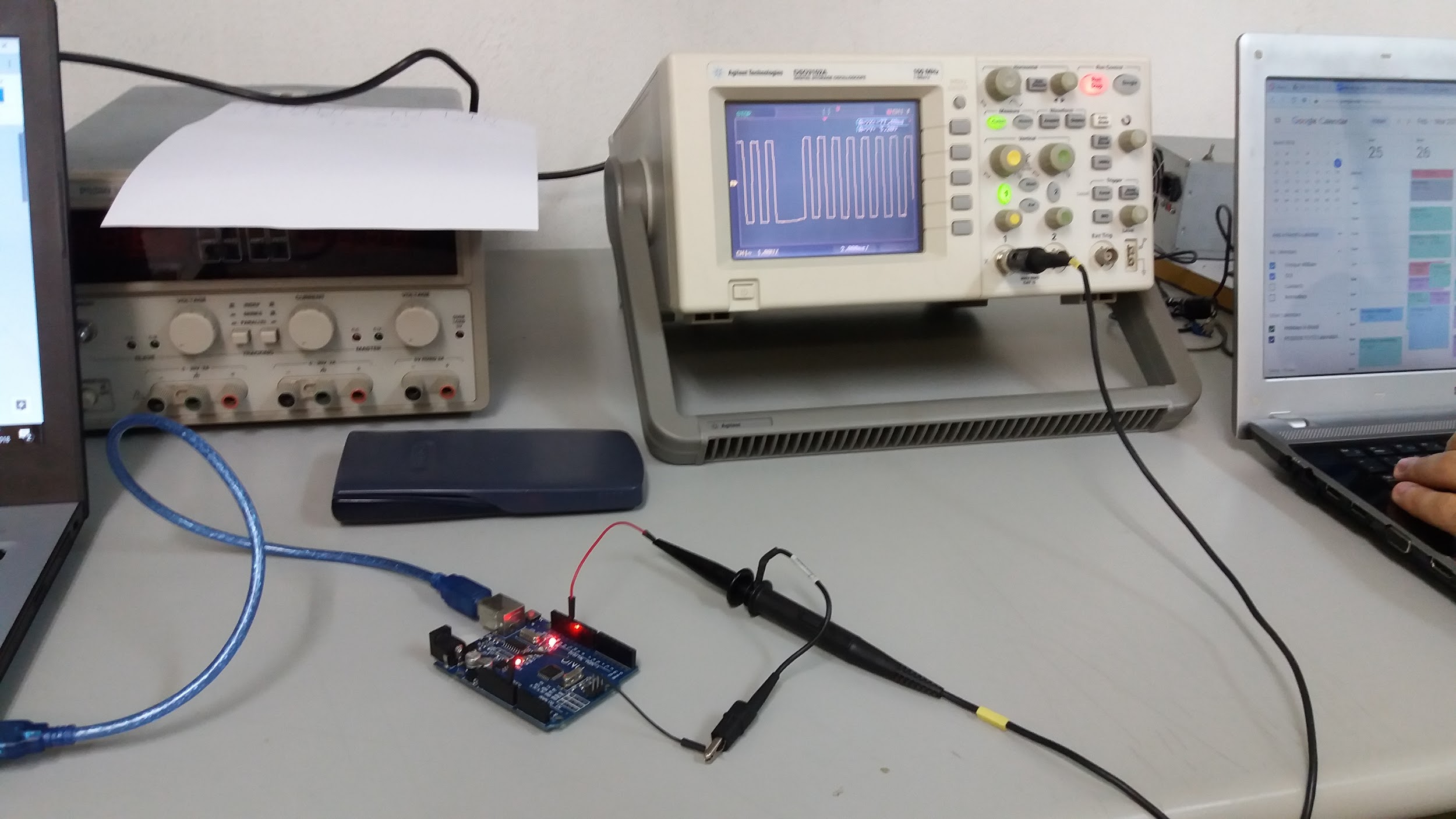
Para a programação do sinal de rotação, o programa definirá o período do sinal de meio dente, a partir de uma rotação de 0 a 6000 rpm definida pelo usuário.

A partir do período de rotação definido para meio dente, o programa deverá alternar o sinal de saída no período *116* vezes, sendo as 4 últimas interações com o dente em zero.

Para isso utilizaremos o Timer 1 do Arduino, programado para levar a uma rotina de interrupção do sistema toda vez que estourá-lo com o período de meio dente.

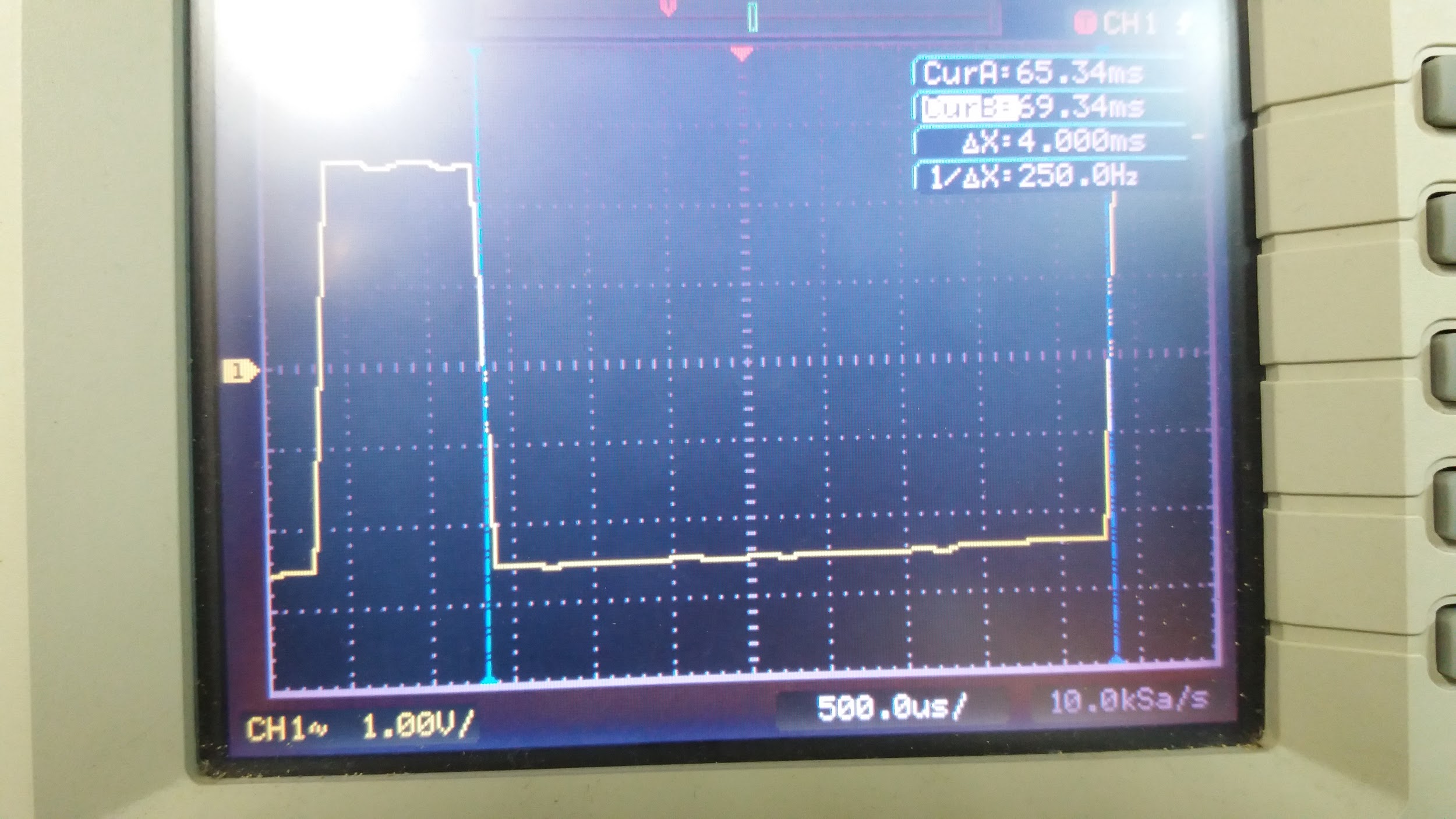
**Testes em bancada e validação do simulador de rotação**

O software presente no anexo 1 tem a função de simular o sinal de rotação a partir da leitura de um sensor do tipo hall em uma roda fônica 60 dentes e 2 falhas. Por meio da aquisição de sinal por um osciloscópio demonstrada na figura abaixo, foram medidos os períodos do sinal de um dente, do sinal de uma falha e o período entre dentes para a validação do software de simulação.

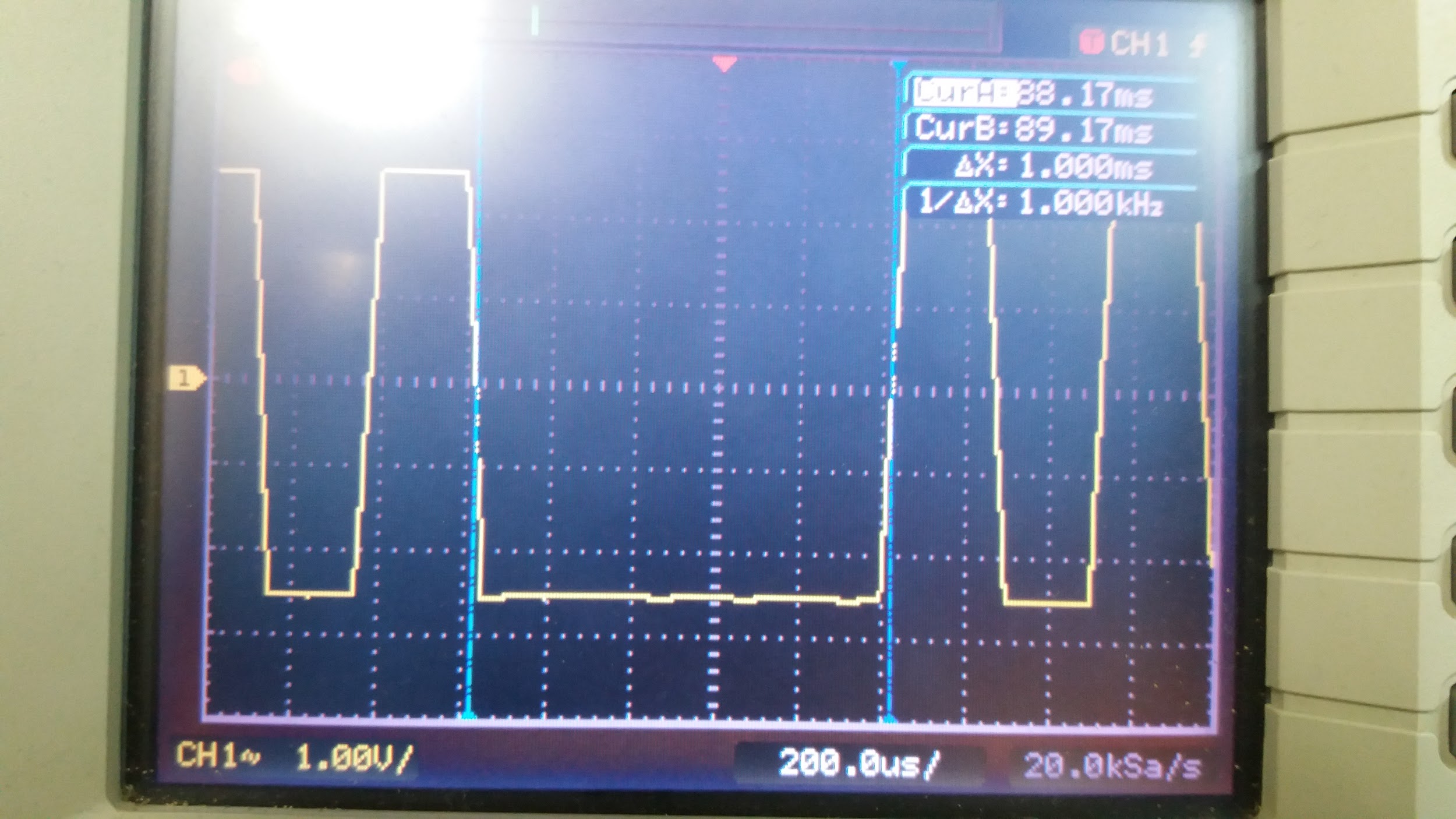
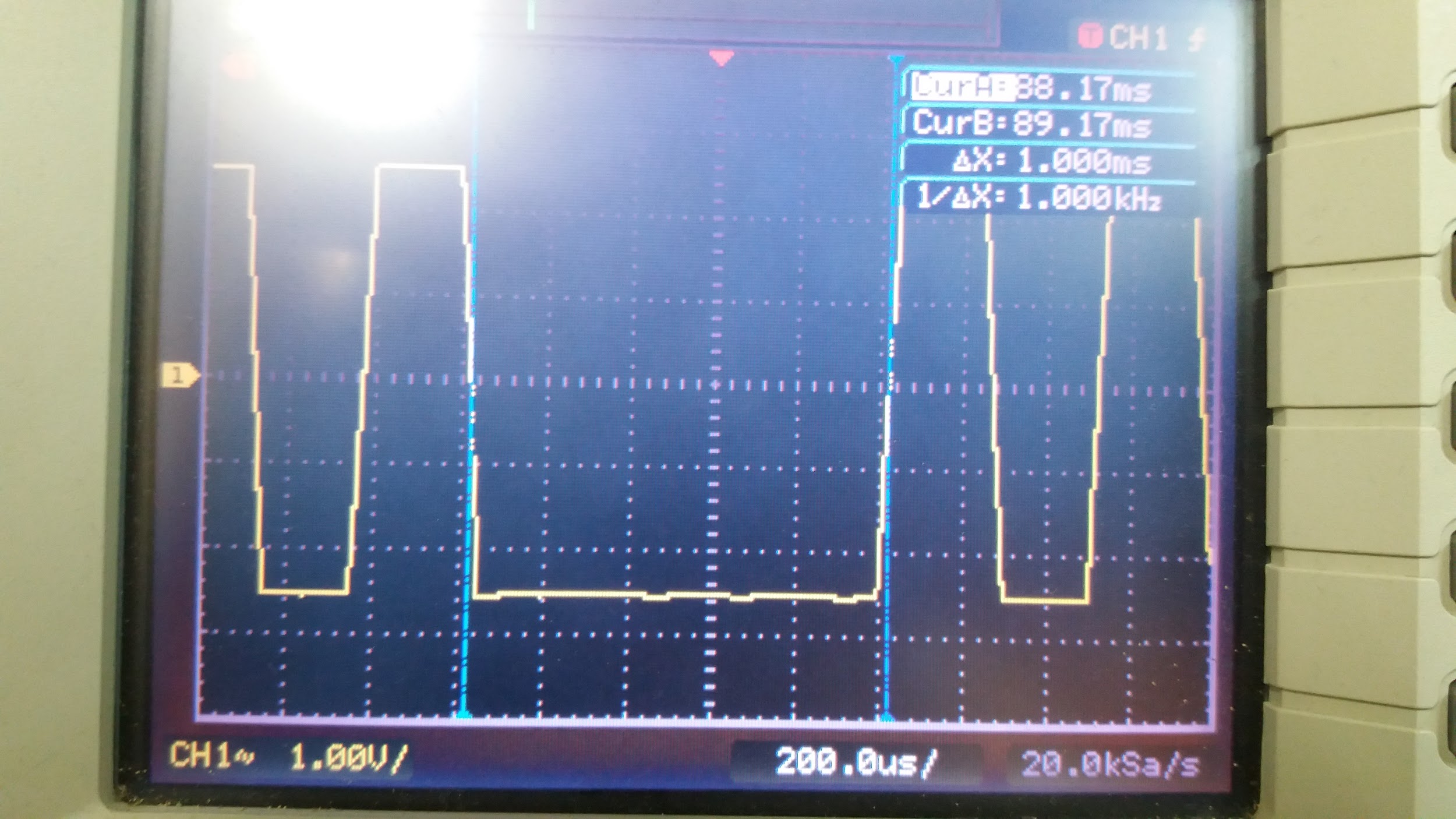


Para simular toda a faixa de operação do motor, medimos a saída do programa para 3 saídas de Rotação: e , obtendo os seguintes resultados:

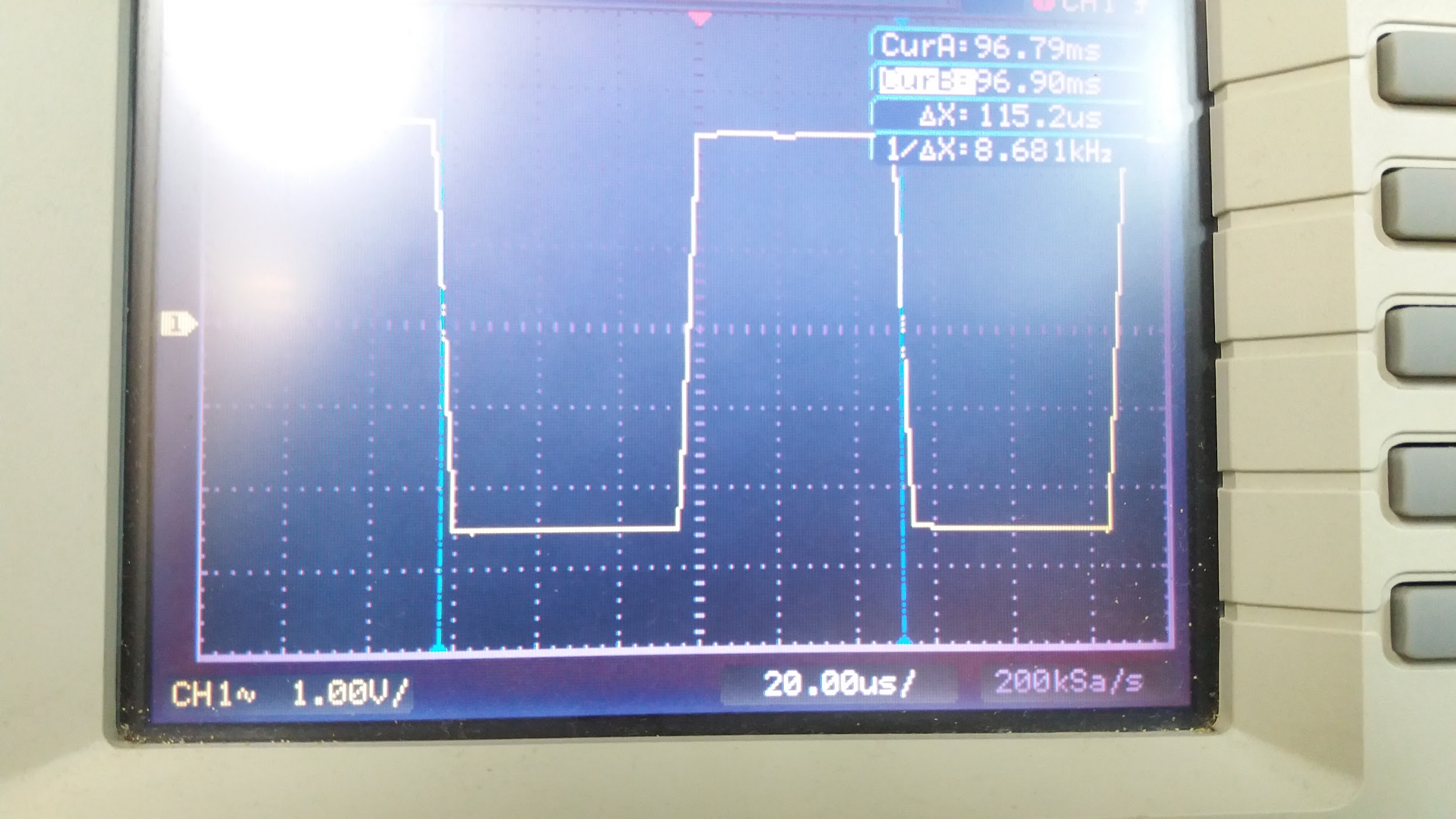
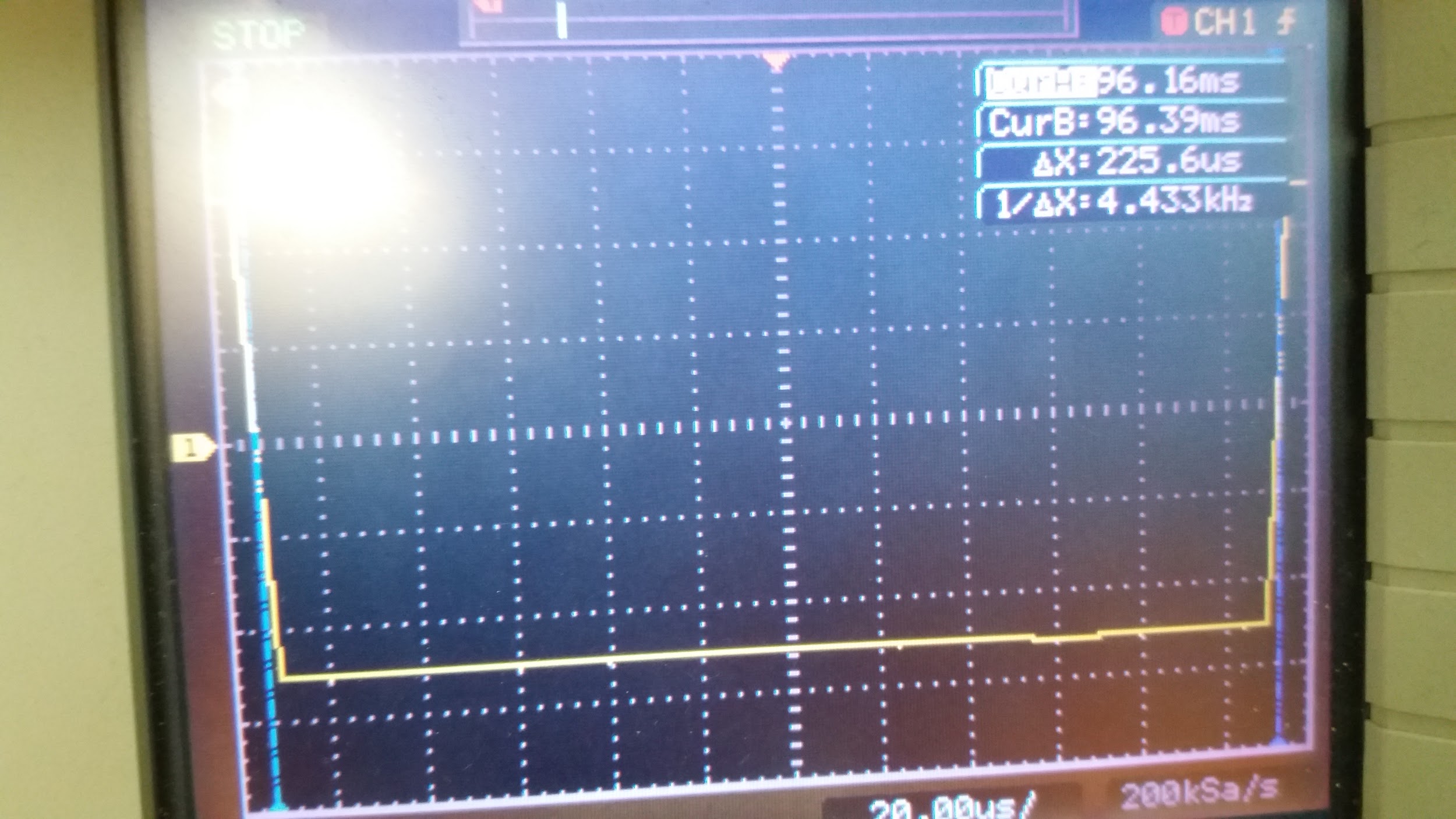
;

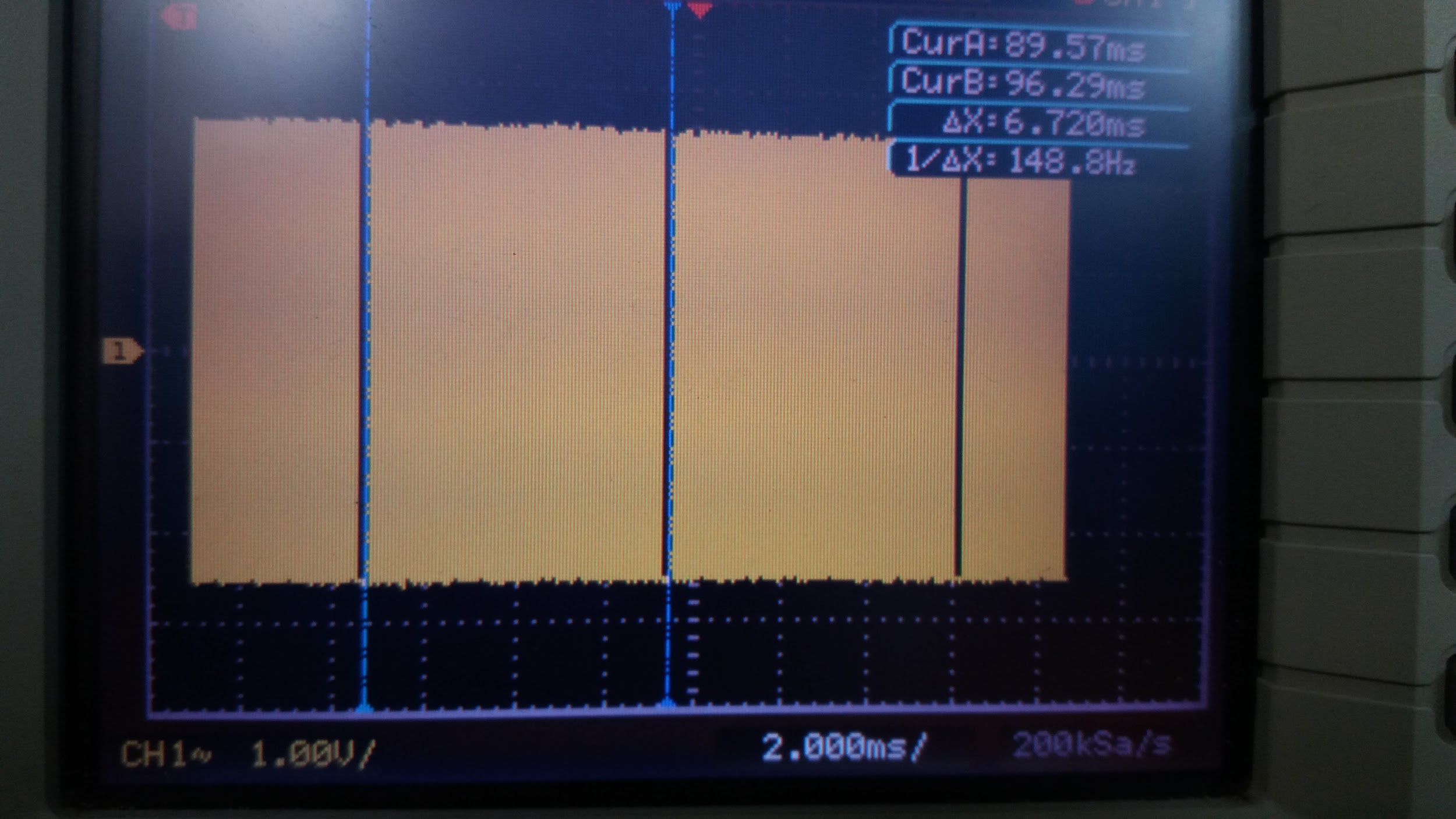


;



**;**





**Simulador de Fase:**

Adicionalmente ao simulador de rotação, foi desenvolvido um simulador do sinal de fase do eixo do comando de válvulas, para que assim possa-se testar os sinais de ignição e injeção do bloco de sincronismo da ECU desenvolvida.

**Sinal de Fase:** Para entender o sinal de fase, devemos entender primeiro o ciclo de funcionamento do eixo do comando de válvulas de um motor à combustão interna e sua funcionalidade.

O eixo do comando de válvulas é responsável pela abertura e fechamento das válvulas de admissão e exaustão em cada um dos cilindros do motor. Esse eixo roda em uma relação 1:2 em comparação com o virabrequim, de forma que, para cada duas voltas do virabrequim, uma volta é realizada no comando de válvulas. Solidário ao eixo de válvula, existem cames defasados angularmente, de forma que, com o movimento do eixo, os cames empurram as válvulas e proporcionam sua abertura e fechamento nos momentos certos do Ciclo Otto.



O sensor de fase trata-se de um sensor muito similar ao sensor de rotação e normalmente está associado à leitura de uma roda dentada, apesar de ter algumas peculiaridades. Sua principal função é identificar, em relação ao PMS do primeiro cilindro, qual cilindro encontra-se no ciclo de combustão. A utilização desse sensor torna-se essencial para simplificar os cálculos do bloco de sincronismo em identificar qual será o cilindro que deverá injetar combustível.

A imagem abaixo (PEREIRA, 2013) demonstra a aquisição do sinal de fase e rotação em um motor de combustão interna do Ciclo Otto.



Pela imagem, nota-se que a geometria da roda dentada não é simples e, consequentemente, do sinal gerado por ela também não é. Mas um padrão em comum pode ser observado facilmente.

A leitura do sinal de fase ocorre quando o bloco de sincronismo identifica a falha da roda fônica, sendo que, nesse caso, o sinal do sensor de fase sempre inverte na próxima identificação de falha. Essa característica também é fundamental para o software da ECU identificar se houve perda do sinal de fase e corrigir caso necessário.

Dessa forma, mostra-se necessário simular um sinal que inverta de estado binário ao completar a primeira volta na árvore de manivelas. No caso da imagem, quando a ECU detecta a falha e lê um sinal de fase como “1”, sabe que nessa volta da árvore de manivelas, injetará sequencialmente nos cilindros 1(PMS no dente 14) e 3 (PMS no dente 44). Ao ler “0” na detecção de falha, o sistema injetará sequencialmente nos cilindros 4 e 2.

A partir dessas informações, foi desenvolvido o software presente no Anexo 2, de forma a englobar a simulação de rotação já discutida em conjunto com o sinal de fase implementado.

**Testes em bancada de Rotação e Fase:**



**2000 rpm**



;

**Conclusões e Comentários:**

**Simulador de Rotação**

Foi desenvolvido um software que simula o sinal de rotação advindo de um sensor do tipo hall em uma unidade de gerenciamento eletrônico de motores do ciclo otto. O erro máximo na pior faixa de operação (Altas velocidades de rotação) é:

**Simulador de Rotação e Fase**

**Anexos:**

**Anexo 1 - Software de simulação de rotação - Arduino UNO IDE**

**Obs: Antes de utilizar, instale a biblioteca TimerOne**

|  |
| --- |
| /\* Programa que simula o sinal vindo de uma roda fônica de 60 dentes e duas falhas Defina a constante RPM com um valor entre 0 rpm e 9000 rpm \*/ #include <TimerOne.h> #include <math.h>  #define RPM 2000  **const** **int** rodaFonica = 13; **volatile** **unsigned** **int** meio\_dente =0; **volatile** **bool** estado = 0; **double** periodo\_da\_volta;  **long** Periodo\_de\_Meio\_Dente(); **long** Periodo\_da\_Falha(); **void** timer1\_ISR();  **void** setup() {  Serial.begin(115200);  pinMode(rodaFonica, **OUTPUT**);  digitalWrite(rodaFonica, **LOW**);  Timer1.initialize(1000); //Seta um tempo de 1000 us no timer 1 do arduino  Timer1.attachInterrupt(timer1\_ISR,Periodo\_de\_Meio\_Dente()); // Ativa interrupção com período calculado para meio dente na rotação especificada, chamando uma rotina de interrupção no estouro } **void** loop() { } **long** Periodo\_de\_Meio\_Dente(){  periodo\_da\_volta = 60000000.0 / (**double**) RPM; // Seta o período em us de uma volta completa do virabrequim  return((**long**) periodo\_da\_volta / 120); // Retorna o período de meio dente de uma roda fônica 60 dentes duas falhas } **void** timer1\_ISR(){ // Rotina de tratamento de Interrupção por Estouro do Timer  Timer1.setPeriod(Periodo\_de\_Meio\_Dente());  meio\_dente++;  estado = !estado;  if(meio\_dente <116)  digitalWrite(rodaFonica, estado);  else{  if(meio\_dente == 116){ //Dá um pulso em 1 equivalente à largura de um dente completo da roda-fônica  Timer1.setPeriod(4\*Periodo\_de\_Meio\_Dente());  estado = 0;  digitalWrite(rodaFonica,estado);  meio\_dente = 0;  }  } } |

**Anexo 2 - Software de simulação de rotação e fase - Arduino UNO IDE**

**Obs: Antes de utilizar, instale a biblioteca TimerOne**

|  |
| --- |
| /\*Programa que simula o sinal vindo de uma roda fônica de 60 dentes e duas falhas e da leitura de uma roda dentada do comando de válvulas (Sensor de fase) \* Defina a constante RPM com um valor entre 0 rpm e 6000 rpm \*/ #include <TimerOne.h> #include <math.h>  #define RPM 8650  **const** **int** rodaFonica = 13; **const** **int** sensorDeFase = 12; **volatile** **unsigned** **int** meio\_dente =0; **volatile** **bool** estadoRot = 0; **volatile** **bool** estadoFase = 0; **double** periodo\_da\_volta;  long periodo\_meio\_dente;  **long** Periodo\_de\_Meio\_Dente(); **long** Periodo\_da\_Falha(); **void** timer1\_ISR();    **void** setup() {  pinMode(rodaFonica, **OUTPUT**);  pinMode(sensorDeFase, **OUTPUT**);  digitalWrite(rodaFonica, **LOW**);  digitalWrite(sensorDeFase, **LOW**);  periodo\_meio\_dente = Periodo\_de\_Meio\_Dente();  Timer1.initialize(1000); //Seta um tempo de 1000 us no timer 1 do arduino  Timer1.attachInterrupt(timer1\_ISR,Periodo\_de\_Meio\_Dente()); // Ativa interrupção com período calculado para meio dente na rotação especificada, chamando uma rotina de interrupção no estouro } **void** loop() { } **long** Periodo\_de\_Meio\_Dente(){  periodo\_da\_volta = 60000000.0 / (**double**) RPM; // Seta o período em us de uma volta completa do virabrequim  return((**long**) periodo\_da\_volta / 120); // Retorna o período de meio dente de uma roda fônica 60 dentes duas falhas }  **void timer1\_ISR(){ // Rotina de tratamento de Interrupção por Estouro do Timer**    **if(meio\_dente <116){**  **Timer1.setPeriod(periodo\_meio\_dente);**  **estadoRot = !estadoRot;**  **digitalWrite(rodaFonica, estadoRot);**  **if(meio\_dente == 60){**  **estadoFase = !estadoFase;**  **digitalWrite(sensorDeFase, estadoFase);**  **}**  **meio\_dente++;**  **}**    **else**  **if(meio\_dente == 116){ //Dá um pulso em 1 equivalente à largura de um dente completo da roda-fônica**  **Timer1.setPeriod(4\*periodo\_meio\_dente);**  **estadoRot = 0;**  **digitalWrite(rodaFonica,estadoRot);**  **meio\_dente = 0;**  **}**  **}** |