

**ИНСТИТУТ ЗА МАТЕМАТИКУ И ИНФОРМАТИКУ**

**ПРИРОДНО-МАТЕМАТИЧКИ ФАКУЛТЕТ УНИВЕРЗИТЕТА У КРАГУЈЕВЦУ**

**Пројекат**

**Предмет: Микропроцесорски системи**

**Тема: Систем за троосно мерење убрзања**

**Студент: Професор:**

**Стефан Петровић 86/2017 др Александар Пеулић**

Садржај

[**Увод** 2](#_Toc112921885)

[**Протеус** 3](#_Toc112921886)

[**STM32CubeIDE** 4](#_Toc112921887)

[Функције 4](#_Toc112921888)

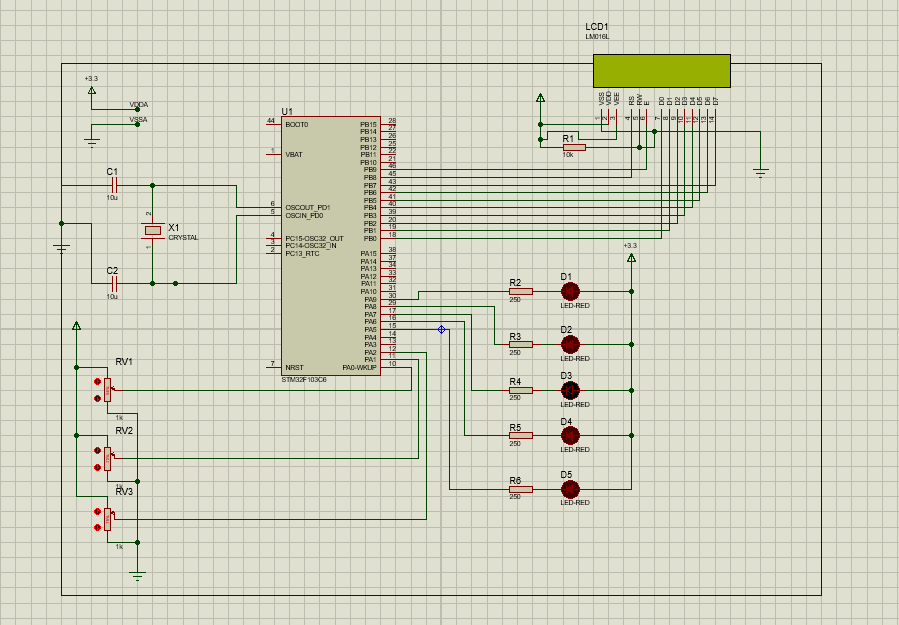
[Main() 6](#_Toc112921889)

# 

# **Увод**

Циљ пројекта је реализација система који омогућава троосно мерење убрзања и њихов приказ.

# **Протеус**



На слици је приказана шема пројекта,као замена за хардверску реализацију пројекта. Уместо потенциометара који симулирају вредности аналогних сигнала који се мере, у реалној хардверској представи користили бисмо одговарајуће сензоре који мере убрзање.

Три потенциометара (пинови 0, 1, 2 порта А) дају одговарајући напон за мерење вредности убрзања на осама X, Y, Z.

Прва диода D1 светли на почетку израчунавања, даје знак да започиње израчунавање. Друга диода D2 светли када се израчунава вредност за Х осу и када се резултат приказује на LD-у, затим се гаси. Затим се пали трећа диода D3, тада се израчунава вредност за Y осу и резулта се приказује на LD-у, и затим се диода гаси. Четврта диода D4 светли када се израчунава вредност за Z осу и исписује резултат на LD-у. Пета диода светсли када се рачуна убрзање и резултат убрзања се приказује на LD-у.

LD је са контролером повезан преко пинова 0-9 порта Б и на њој се приказују резултати израчунавања.

# **STM32CubeIDE**

## Функције

**void** **LCD\_init**()

{

LCD(0x38, 0);

LCD(0x0C, 0);

LCD(0x06, 0);

LCD(0x01, 0);

LCD(0x80, 0);

}

Функција ***LCD\_init()*** дефинисе почетно стање LD-a.

**void** **LCD**(uint8\_t val\_1, uint8\_t cmd)

{

uint8\_t data1;

HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_8, cmd);

data1 = val\_1 & 0x01;

HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_0, data1);

data1 = (val\_1 >> 1) & 0x01;

HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_1, data1);

data1 = (val\_1 >> 2) & 0x01;

HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_2, data1);

data1 = (val\_1 >> 3) & 0x01;

HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_3, data1);

data1 = (val\_1 >> 4) & 0x01;

HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_4, data1);

data1 = (val\_1 >> 5) & 0x01;

HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_5, data1);

data1 = (val\_1 >> 6) & 0x01;

HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_6, data1);

data1 = (val\_1 >> 7) & 0x01;

HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_7, data1);

}

Функција ***LCD()*** исписује послати карактер. Уколико се као вредност аргумента цмд добије 0, послати карактер ће бити посматран као команда, уколико се добије 1, онда ће бити исписана вредхост на LCD-у.

**void** **lcd\_puts**(**const** **char** \* s)

{

**while**(\*s)

LCD(\*s++,1);

}

Функција ***lcd\_puts(***) исписује прослеђену реч на LCD–у.

**void** **ispis\_broja**(**float** broj)

{

**int** ceo;

**float** ost;

**int** ostCeo=0;

**int** i=0;

**char** buffer[10];

ceo = broj;

ost = broj-ceo;

ostCeo = ost\*1000000;

ostCeo=**abs**(ostCeo);

**sprintf**(buffer,"%d",ceo);

**while**(buffer[i])

{

LCD(buffer[i],1);

i++;

}

LCD('.',1);

**sprintf**(buffer,"%d",ostCeo);

i=0;

**while**(buffer[i])

{

LCD(buffer[i],1);

i++;

}

}

Функција ***ispis\_broja()*** исписује прослеђени број на екрану LCD-а. Прослеђени број дели на цео и реалан део и сваку цифру посебно исписује на LCD-у.

## Main()

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_5, *GPIO\_PIN\_RESET*);

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_6, *GPIO\_PIN\_RESET*);

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_7, *GPIO\_PIN\_RESET*);

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_8, *GPIO\_PIN\_RESET*);

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_9, *GPIO\_PIN\_RESET*);

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_9, *GPIO\_PIN\_SET*);

LCD\_init();

lcd\_puts("POCINJE\_RACUNANJE");

HAL\_Delay(100);

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_9, *GPIO\_PIN\_RESET*);

LCD\_init();

На почетску су све диоде угашене и исписује се порука да почиње рачунање.

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_8, *GPIO\_PIN\_SET*);

HAL\_ADC\_Start(&hadc1);

HAL\_ADC\_PollForConversion(&hadc1, 1000);

vrednost=HAL\_ADC\_GetValue(&hadc1);

aX=(vrednost\*3.3)/4096+6.1;

LCD\_init();

lcd\_puts("X:");

ispis\_broja(aX);

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_8, *GPIO\_PIN\_RESET*);

Пали се диода и даје ѕнак да је започета конверзија. Рачуна се вредност убрзања за Х осу попоћу одговарајуће формуле где сеаналогни сигнал који потенциометар даје претвара у реалан број. Дати резултат се исписује на екрану. Затим се диода гаси и брисе све са екрана.

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_7, *GPIO\_PIN\_SET*);

HAL\_ADC\_PollForConversion(&hadc1, 1000);

vrednost=HAL\_ADC\_GetValue(&hadc1);

aY=(vrednost\*3.3)/4096+0.5;

LCD\_init();

lcd\_puts("Y:");

ispis\_broja(aY);

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_7, *GPIO\_PIN\_RESET*);

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_6, *GPIO\_PIN\_SET*);

HAL\_ADC\_PollForConversion(&hadc1, 1000);

vrednost=HAL\_ADC\_GetValue(&hadc1);

aZ=(vrednost\*3.3)/4096+2.7;

LCD\_init();

lcd\_puts("Z:");

ispis\_broja(aZ);

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_6, *GPIO\_PIN\_RESET*);

Исти поступак, као за израчунавање вредности за Х осу, користи се за израчунавање вредности за Y I Z осу.

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_5, *GPIO\_PIN\_SET*);

res=**sqrt**(aX\*aX+aY\*aY+aZ\*aZ);

LCD\_init();

lcd\_puts("UBRZANJE JE:");

LCD\_init();

ispis\_broja(res);

lcd\_puts("m/s^2");

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_5, *GPIO\_PIN\_RESET*);

На крају се пали диода, израчунавамо вредност убрзања помоћу дате формуле и приказујемо на екрану. Диода се гаси и поново започињемо рачунање.