

HƯỚNG DẪN ĐIỀU KHIỂN VẬN TỐC ĐỘNG CƠ DC SERVO

Tạo project

Mở **Cube MX** -> chọn **ACCESS TO MCU SELECTOR** -> trong ô tìm kiếm, nhập **STM32F103RCT6** -> chọn vi xử lý tương ứng

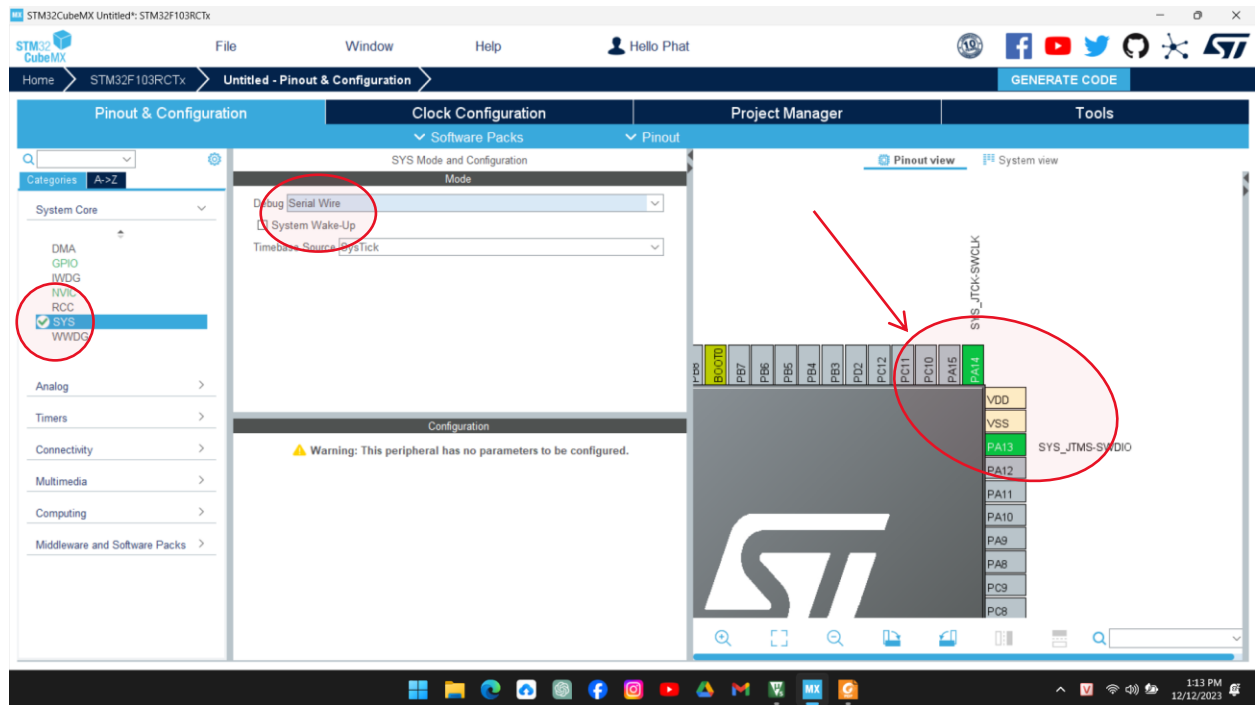
1

2

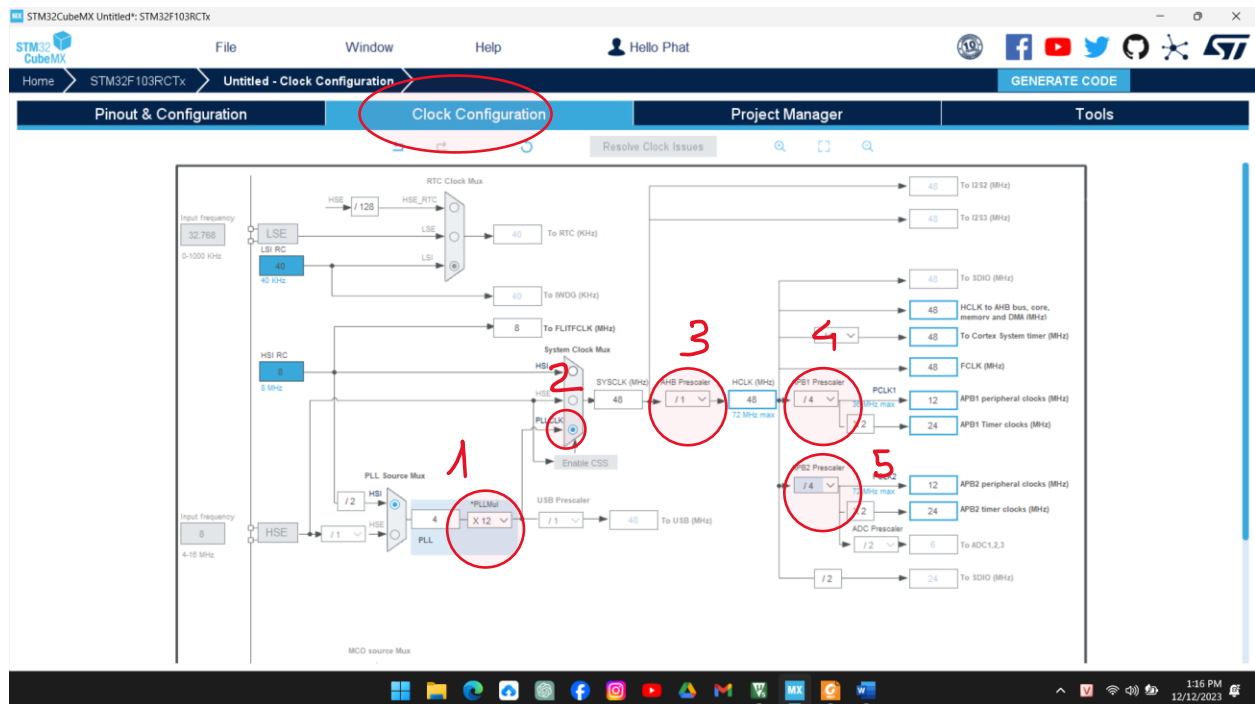
MCUs/MPUs List: 2 items	Part No	Reference	Marketing St...	Unit Price for 10KU [...]	Board	Package	Flash	RAM	IO	Freque...
Commercial Part	STM32F103RCT6	STM32F103RCTx	Active	4.0035		LQFP 64 10x10x1.4 mm	256 kBytes	48 kBytes	51	72 MHz
	STM32F103RCT6TR	STM32F103RCTx	Active	4.0035		LQFP 64 10x10x1.4 mm	256 kBytes	48 kBytes	51	72 MHz

Cấu hình vi xử lý

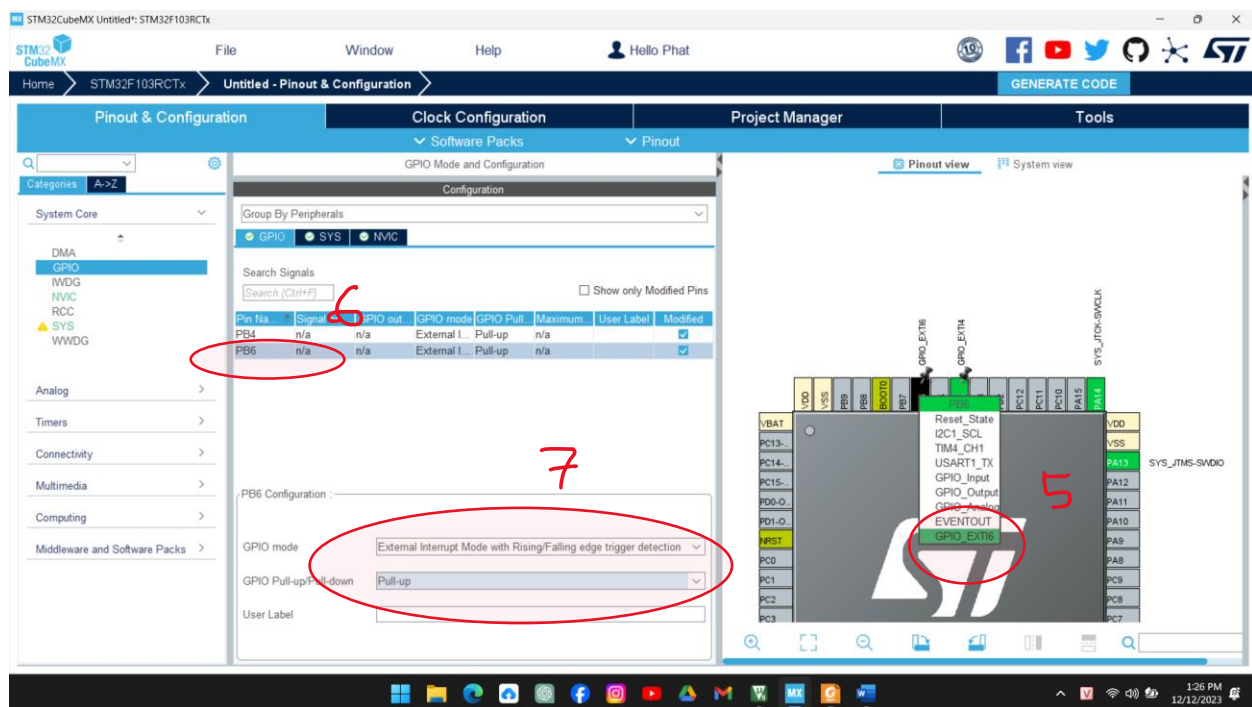
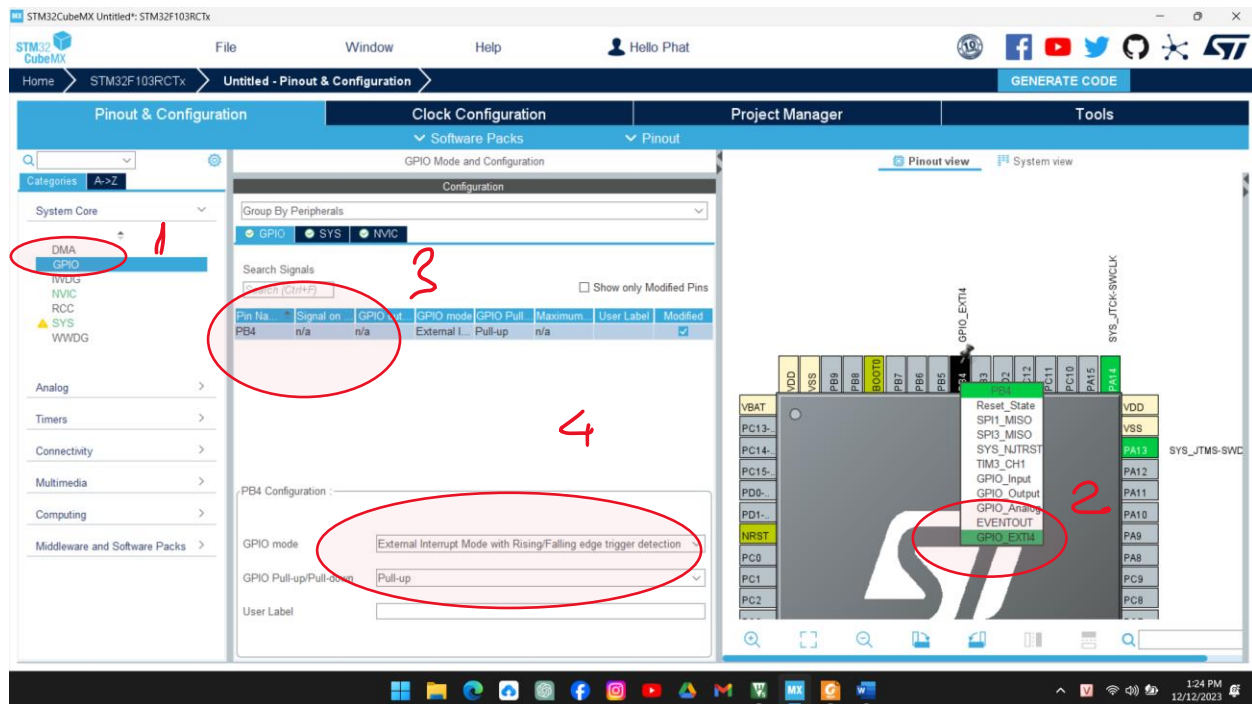
Bước 1: Cài debug serial wire

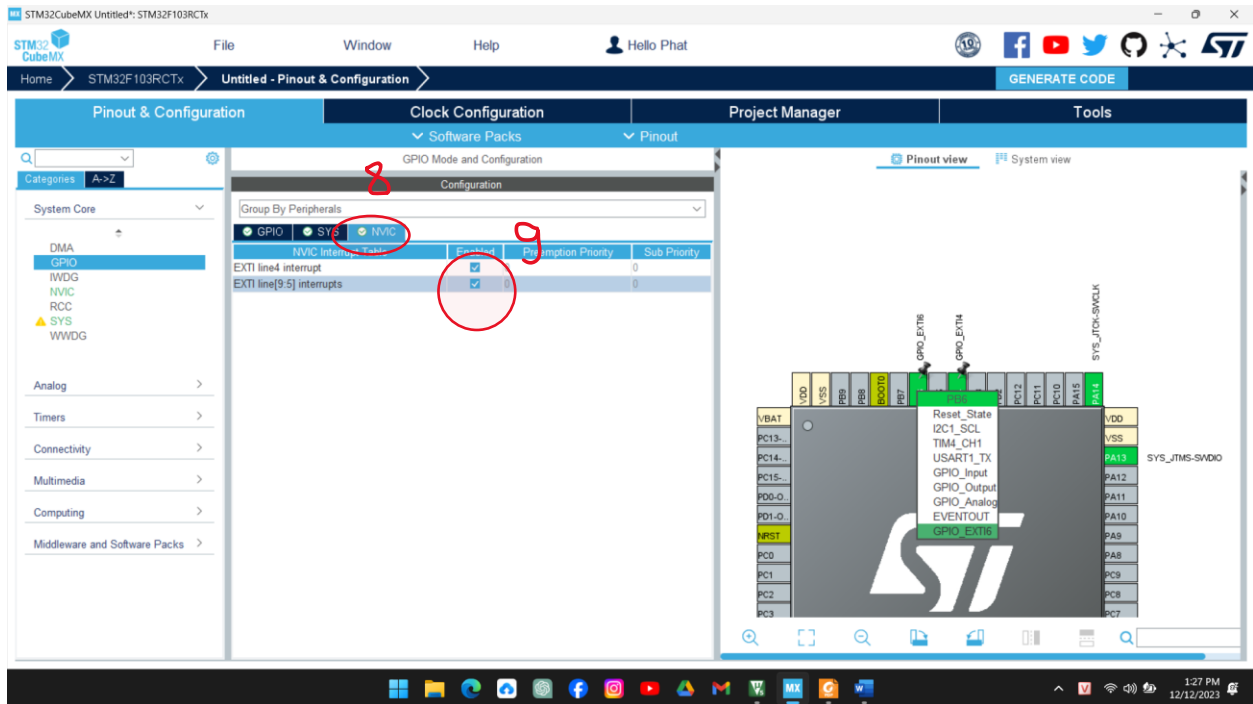


Bước 2: Cấu hình clock cho hệ thống

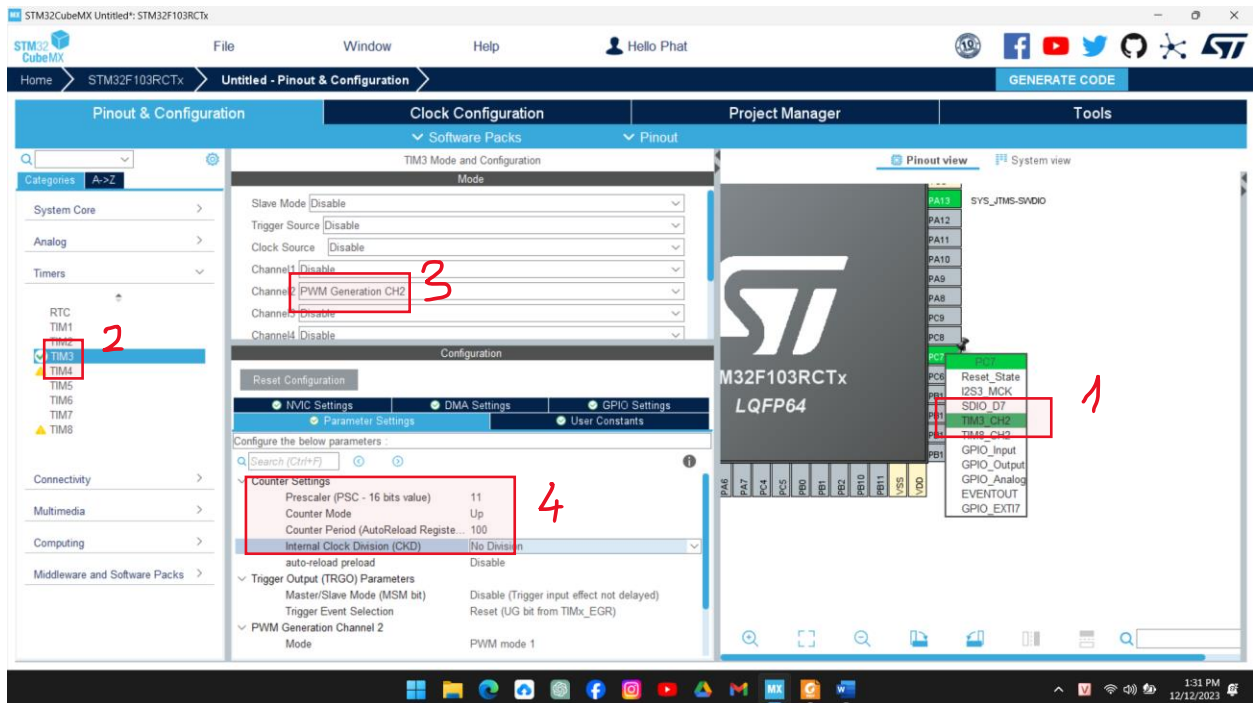


Bước 3: Cấu hình hai chân ngắt để đọc encoder

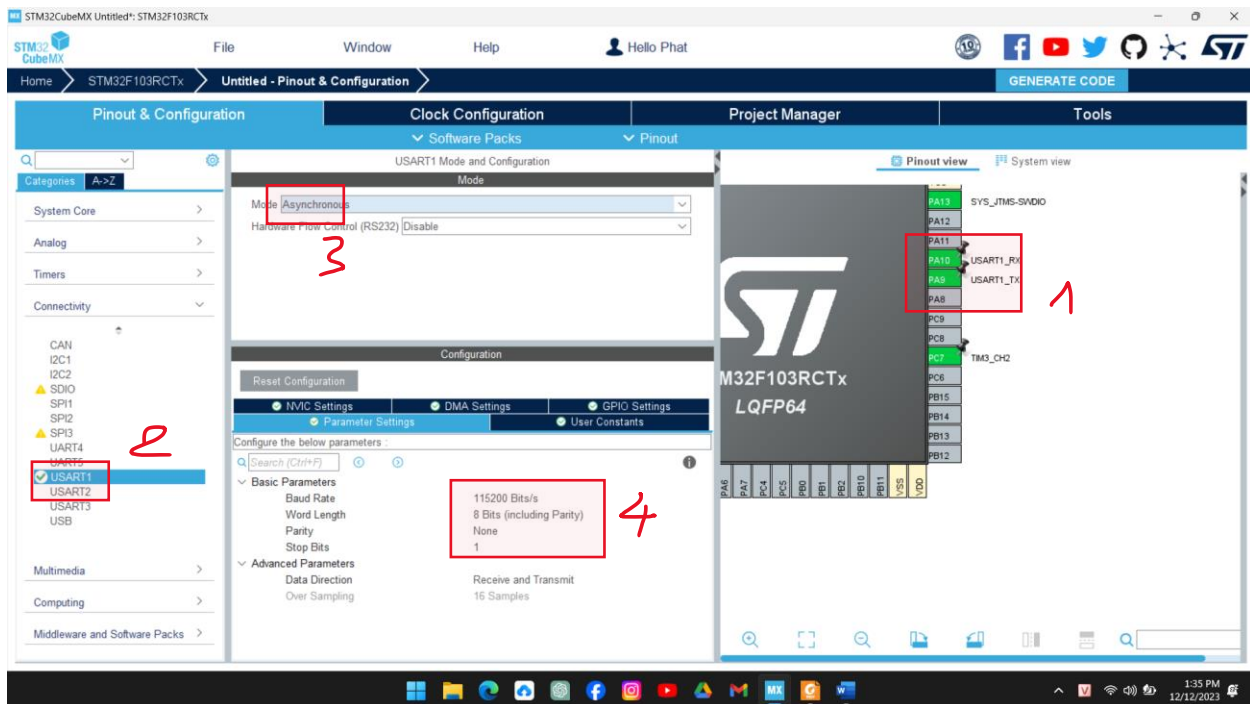




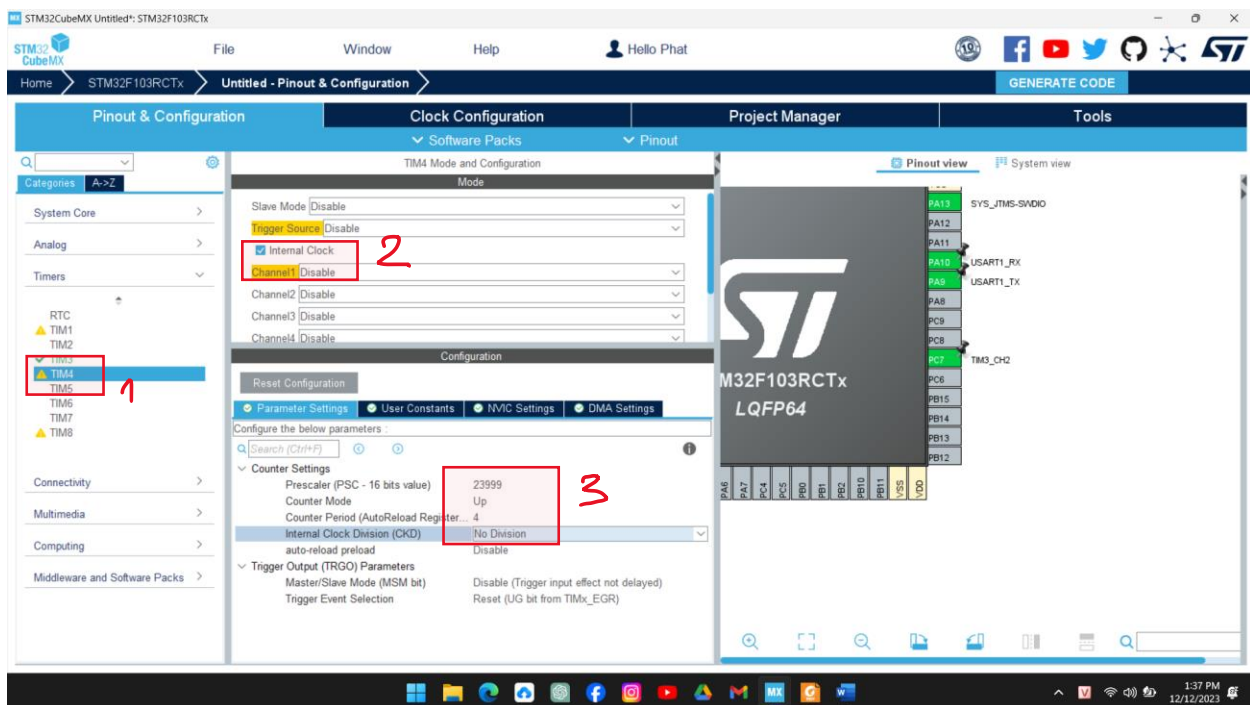
Bước 4: Cấu hình chân xuất xung PWM

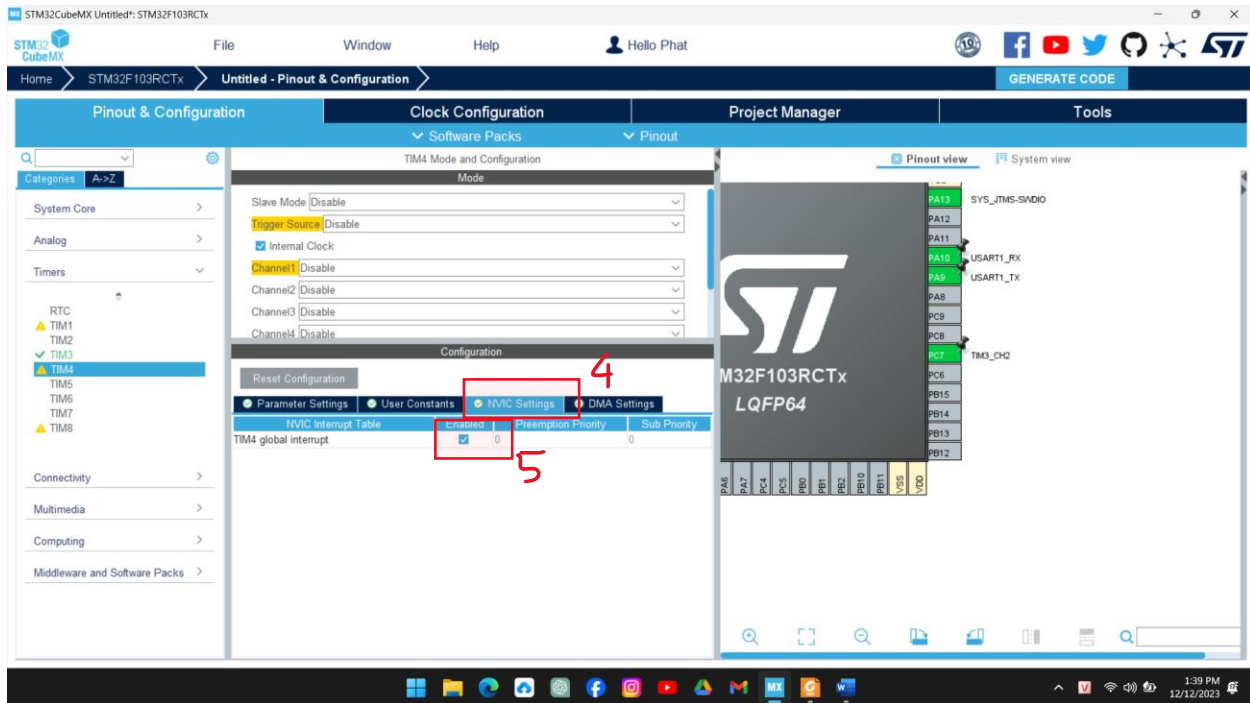


Bước 5: Cấu hình chân giao tiếp UART

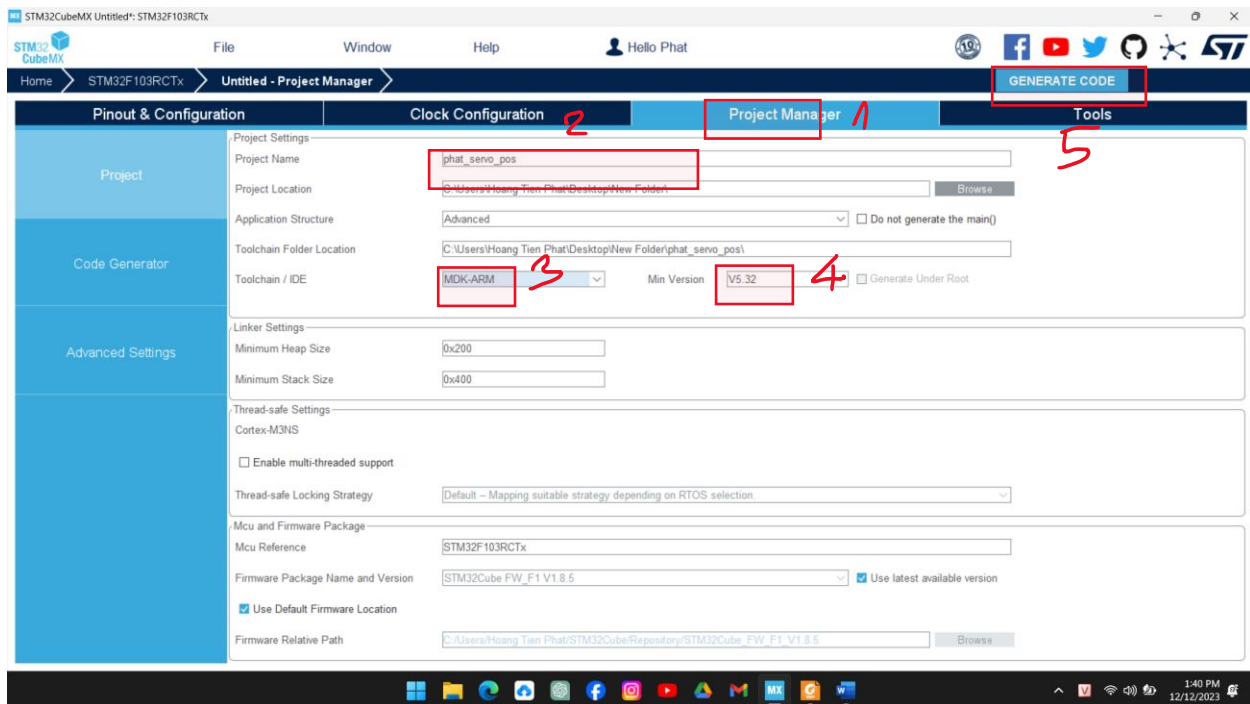


Bước 6: Cấu hình ngắt timer4 5ms để tính vận tốc





Bước 7: Sinh code



Khai báo các thư viện, biến và hàm

```
1  #include "main.h"
2  #include "string.h"
3  #include "stdio.h"
4  #include "stdlib.h"
5  #include "stdbool.h"
6
7  #define pi 3.1415
8  #define p2r pi/2000
9  #define Kp 33.54
10 #define Kb 25.52
11 #define alpha 0.1
12
13 TIM_HandleTypeDef htim2;
14 TIM_HandleTypeDef htim3;
15 TIM_HandleTypeDef htim4;
16 TIM_HandleTypeDef htim5;
17
18 UART_HandleTypeDef huart1;
19
20 /* USER CODE BEGIN PV */
21 int32_t PosCnt, Cnttmp, speed;
22 int16_t CountValue=0, RealVel, DesiredSpeed, HILIM, LOLIM;
23 uint16_t AngPos1, AngPos0, CntVel;
24 uint8_t PreviousState, pwm, Speedmode, tick=0;
25 bool run=false, dir;
26 float CurPos=0, DesiredPos, CurVel;
27 char Rx_inx, Rx_Buffer[20], Rx_data[2];
28 float sample_time = 0.005;
29 float gain, Ki, Kd;
30
31 void SystemClock_Config(void);
32 static void MX_GPIO_Init(void);
33 static void MX_TIM2_Init(void);
34 static void MX_TIM3_Init(void);
35 static void MX_USART1_UART_Init(void);
36 static void MX_TIM4_Init(void);
37 static void MX_TIM5_Init(void);
38
39 void HAL_TIM_MspPostInit(TIM_HandleTypeDef *htim);
40 int SetVelLow(float CurrentPos, float Pos);
41 int SetVelMid(float CurrentPos, float Pos, float CurrentVel);
42 int SetVelHigh(float CurrentPos, float Pos, float CurrentVel);
43 int PIDCtrl(float Desired, float Current, float p_coef, float i_coef, float d_coef);
44
45 #ifdef __GNUC__
46 #define PUTCHAR_PROTOTYPE int __io_putchar(int ch)
47 #else
48 #define PUTCHAR_PROTOTYPE int fputc(int ch, FILE *f)
49 #define GETCHAR_PROTOTYPE int fgetc(FILE *f)
50 #endif
51 PUTCHAR_PROTOTYPE
52 {
53     HAL_UART_Transmit(&huart1, (uint8_t*)&ch, 1, 100);
54     return ch;
55 }
```

công thức tính ở gần cuối tài liệu

Hàm UART1 (thầy cho sẵn rồi)

```
58 void HAL_UART_RxCpltCallback(UART_HandleTypeDef *huart) // Ham ngat Uart
59 {
60     uint8_t i;
61     if(huart->Instance == USART1) //uart1
62     {
63         if(Rx_indx==0) {for (i=0;i<20;i++) Rx_Buffer[i] = 0;}
64
65         switch(Rx_data[0]) {
66             /* dung dong co */
67             case 'e':
68                 run = false;
69                 break;
70
71             /* dong co chay */
72             case 'r':
73                 run = true;
74                 break;
75             case 'b':
76                 // reset();
77                 break;
78             case 's':
79                 DesiredPos = atoi(Rx_Buffer);
80                 memset(Rx_Buffer, 0, sizeof(Rx_Buffer));
81                 Rx_indx = 0;
82                 break;
83             case 'v':
84                 DesiredSpeed = atoi(Rx_Buffer);
85                 memset(Rx_Buffer, 0, sizeof(Rx_Buffer));
86                 Rx_indx = 0;
87                 break;
88             case '0':
89             case '1':
90             case '2':
91             case '3':
92             case '4':
93             case '5':
94             case '6':
95             case '7':
96             case '8':
97             case '9':
98             case '.':
99             case '-':
100                 Rx_Buffer[Rx_indx++] |= Rx_data[0];
101                 break;
102             default:
103                 break;
104         }
105         HAL_UART_Receive_IT(&huart1, (uint8_t*)Rx_data, 1);
106     }
107 }
```


Hàm ngắt ngoài PB6 đọc encoder CHB (thầy cho sẵn rồi)

```
109 void EXTI9_5_IRQHandler(void) // doc encoder
110 {
111     /* USER CODE BEGIN EXTI9_5_IRQn 0 */
112     unsigned char State0;
113     State0 = (State0<<1) | HAL_GPIO_ReadPin(GPIOB, GPIO_PIN_4);
114     State0 = (State0<<1) | HAL_GPIO_ReadPin(GPIOB, GPIO_PIN_6);
115     State0 = State0&0x03;
116     switch (State0) {
117         case 0:
118             if(PreviousState==1) CountValue++;
119             else CountValue--;
120             break;
121         case 1:
122             if(PreviousState==3) CountValue++;
123             else CountValue--;
124             break;
125         case 2:
126             if(PreviousState==0) CountValue++;
127             else CountValue--;
128             break;
129         case 3:
130             if(PreviousState==2) CountValue++;
131             else CountValue--;
132             break;
133     }
134     PreviousState = State0;
135     CntVel++;
136     if (CountValue>2000) {
137         CountValue = 0;
138         PosCnt++;
139     }
140     else if (CountValue<-2000) {
141         CountValue = 0;
142         PosCnt--;
143     }
144     /* USER CODE END EXTI4_IRQn 0 */
145     HAL_GPIO_EXTI_IRQHandler(GPIO_PIN_6);
146     /* USER CODE BEGIN EXTI4_IRQn 1 */
147 }
```

≥ 4000

≤ -4000

thầy sửa sẵn rồi

Hàm ngắt ngoài PB4 đọc encoder CHA (thầy cho sẵn rồi)

```
150 void EXTI4_IRQHandler(void) // doc encoder
151 {
152     // CHANNEL A
153     unsigned char Statel;
154     Statel = (Statel<<1) | HAL_GPIO_ReadPin(GPIOB, GPIO_PIN_4);
155     Statel = (Statel<<1) | HAL_GPIO_ReadPin(GPIOB, GPIO_PIN_6);
156     Statel = Statel&0x03;
157     switch (Statel) {
158         case 0:
159             if(PreviousState==1) CountValue++;
160             else CountValue--;
161             break;
162         case 1:
163             if(PreviousState==3) CountValue++;
164             else CountValue--;
165             break;
166         case 2:
167             if(PreviousState==0) CountValue++;
168             else CountValue--;
169             break;
170         case 3:
171             if(PreviousState==2) CountValue++;
172             else CountValue--;
173             break;
174     }
175     PreviousState = Statel;
176     CntVel++;
177     if (CountValue>=2000) {
178         CountValue = 0;
179         PosCnt++;
180     }
181     else if (CountValue<=-2000) {
182         CountValue = 0;
183         PosCnt--;
184     }
185     HAL_GPIO_EXTI_IRQHandler(GPIO_PIN_4);
186 }
187
```

≥ 4000

≤ -4000

thầy sửa sẵn rồi

Hàm ngắt TIM4 mỗi 5ms tính vận tốc

```
188 void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim) // ngắt timer 4 tính vận t
189 {
190     if(htim->Instance==TIM4) // ngắt do timer 4 5ms
191     {
192         CurPos = PosCnt*2*pi+CountValue*p2r; // Position calculation
193         Cnttmp = CntVel; // biến lưu tạm thời
194         CntVel = 0;
195         RealVel = Cnttmp*6; //RPM 60/2000x0.005
196         CurVel = Cnttmp*pi/5; //rad/s 2pi/2000x0.005
197         switch(Speedmode) {
198             case 1:
199                 pwm = SetVelLow(CurPos,DesiredPos);
200                 break;
201             case 2:
202                 pwm = SetVelMid(CurPos,DesiredPos,CurVel);
203                 break;
204             case 4:
205                 pwm = SetVelHigh(CurPos,DesiredPos,CurVel);
206                 break;
207         }
208         // dir = 1, CurPos<0
209         if (run==true){
210             // HAL_TIM_SetCompare(&htim2, TIM_CHANNEL_3,0); // dir
211             gain = 0.035; //0.674 Kp
212             Ki = 0.8932; //17.2
213             Kd = 0;
214             pwm = PIDCtrl(DesiredSpeed,RealVel,gain,Ki,Kd);
215             HAL_GPIO_WritePin(GPIOC,GPIO_PIN_3, GPIO_PIN_SET);
216             __HAL_TIM_SetCompare(&htim3,TIM_CHANNEL_2,pwm); // set pwm
217         }
218         else {
219             HAL_GPIO_WritePin(GPIOC,GPIO_PIN_3, GPIO_PIN_RESET);
220             // __HAL_TIM_SetCompare(&htim2, TIM_CHANNEL_3,0); // set pwm
221             __HAL_TIM_SetCompare(&htim3, TIM_CHANNEL_2,pwm); // set pwm
222         }
223         return;
224     }
225     if(htim->Instance==TIM5)
226     {
227         // ut=5*pi*time; // hàm chạy theo thời gian u(t)=5*pi(t); // timer 5
228         // time+=0.01;
229         tick++;
230         if (run==1) {
231             if(DesiredSpeed<=340) Speedmode = 1;
232             else if((DesiredSpeed>340)&&(DesiredSpeed<750)) Speedmode = 2;
233             else if(DesiredSpeed>=750) Speedmode = 4;
234         }
235         if (run==0){
236             pwm = 0;
237             Speedmode = 0;
238         }
239         if ((run==1)&&(tick==5)){
240             tick=0;
241             printf("V%d\r \n",RealVel);
242             //printf("P%f\r \n",CurPos);
243         }
244     }
245 }
246 }
```

thầy sửa sẵn rồi

gain là $K_c=K_p$

cách tính bên dưới

thay pwm thành 100 để vẽ

Ba hàm SetVel (thầy cho sẵn rồi)

```
247 int SetVelLow(float CurrentPos, float Pos)
248 {
249     HILIM=10,LOLIM=0;
250     float uout;
251
252     // STUDENTS ADD CODE FOR SLOW SPEED HERE
253
254     if (uout>HILIM) uout=HILIM;
255     else if (uout<LOLIM) uout=LOLIM;
256     return uout;
257 }
258 int SetVelMid(float CurrentPos, float Pos, float CurrentVel)
259 {
260     HILIM=20,LOLIM=0;
261     float uout;
262
263     // STUDENTS ADD CODES FOR MIDDLE SPEED HERE
264
265     if (uout>HILIM) uout=HILIM;
266     else if (uout<LOLIM) uout=LOLIM;
267     return uout;
268 }
269 int SetVelHigh(float CurrentPos, float Pos, float CurrentVel)
270 {
271     HILIM=99,LOLIM=0;
272     float uout;
273
274     // STUDENTS ADD CODES FOR HIGH SPEED HERE
275
276     if (uout>HILIM) uout=HILIM;
277     else if ( uout<LOLIM) uout=LOLIM;
278
279     return uout;
280 }
```

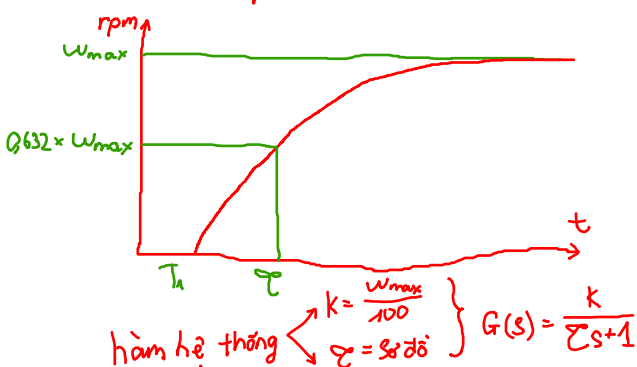
Hàm tính PID

```

282 int PIDCtrl(float Desired, float Current, float p_coef, float i_coef, float d_coef)
283 {
284     static float err_p = 0;
285     static float iterm_p = 0;
286     static float err_sat = 0;
287     static float dterm_f_p = 0;
288
289     float err, err_windup;
290     float pterm, dterm, dterm_f, iterm;
291     float pidterm, pid_sat;
292     int16_t pidout;
293
294     HILIM=100, LOLIM=0;
295
296     // P
297     err = Desired - Current;
298     pterm = p_coef*err;
299     // D
300     dterm = d_coef*(err-err_p)/sample_time;
301     dterm_f = alpha*dterm+(1-alpha)*dterm_f_p;
302     // I
303     err_windup = i_coef*err + Kb*err_sat;
304     iterm = iterm_p + err_windup*sample_time;
305     iterm_p = iterm;
306
307     err_p = err;
308     dterm_f_p = dterm_f;
309     pidterm = pterm + dterm_f + iterm;
310
311     //saturation of PD term
312     if (pidterm>HILIM)
313         pid_sat = HILIM;
314     else if (pidterm<LOLIM)
315         pid_sat = LOLIM;
316     else
317         pid_sat = pidterm;
318
319     err_sat = pid_sat-pidterm; //u non -u --> bao hoa
320     pidout = (int16_t)pid_sat;
321     return pidout;
322 }

```

xuất pwm 100%



Sau khi có T_d, ε

$$T = \varepsilon - T_d$$

Chọn T_c sấp xỉ T (với $T_d = 0,151 \rightarrow T_c = 0,17$)

$$K_c = \frac{T}{k \cdot T_c}$$

$$\varepsilon_i = T$$

$$K_b = \frac{1}{\varepsilon_i}$$

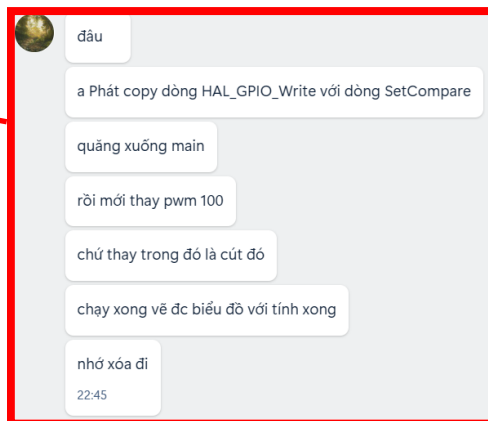
$$K_p = K_c$$

$$K_i = \frac{K_c}{\varepsilon_i}$$

$$K_d = 0$$

Hàm main

```
324 int main(void)
325 {
326     HAL_Init();
327     SystemClock_Config();
328     MX_GPIO_Init();
329     MX_TIM2_Init();
330     MX_TIM3_Init();
331     MX_USART1_UART_Init();
332     MX_TIM4_Init();
333     MX_TIM5_Init();
334
335     HAL_TIM_PWM_Start(&htim2, TIM_CHANNEL_3); // khởi tạo timer 2
336     HAL_TIM_PWM_Start(&htim3, TIM_CHANNEL_2); // khởi tạo timer 3
337     HAL_TIM_Base_Start_IT(&htim4);           // khởi tạo timer 4
338     HAL_TIM_Base_Start_IT(&htim5);           // khởi tạo timer 5
339     HAL_UART_Receive_IT(&huart1, (uint8_t*)Rx_data, 1);
340
341     while (1)
342     {
343     }
344 }
345 }
```



sau khi code xong, chạy debug, add các biến

DesiredSpeed, RealVel, pwm, gain, Ki, Kd, run vào watch1

set các giá trị

Kp, Kb, Ki, Kd

cho run = true để bắt đầu chạy
nhập giá trị mong muốn cho DesiredSpeed
xem giá trị hiện tại ở biến RealVel

