

بسم الله الرحمن الرحيم

علیرضا اینالو هستم

با شماره دانشجویی ۳۹۹۱۲۳۴۰۱۱۹۱۱۸

استاد راهنمای بنده دکتر صفری هستن

نام پروژه خط تولید می باشد

سال تحصیلی ۰۳-۰۲

هدف از این چند صفحه ارائه یه گزارش از عملکرد پروژه انجام شده برای درس سیستم های ریز پردازنده می باشد که با توجه به تایید استاد راهنما، جناب دکتر صفری صورت گرفته است.

در ادامه می خواهم یه کلیتی درباره کار کرد و هدف این پروژه خدمتون ارائه کنم و سپس بخش های مختلف برنامه رو براتون مورد شرح و بررسی قرار بدم:

- انتخاب پلتفرم برای انجام این پروژه
- آماده سازی محیط لنجام پروژه
- شبیه سازی سنسور های مورد نیاز
- توضیح بخش های عملکرد سیستم:
- کتاب خونه های مورد استفاده
- تسک های مورد استفاده در سیستم
- ارتباط تسک ها

قبل از شروع از اینکه وقت خود را برای مطالعه این گزارش می گزارین ازتون ممنونم.

### هدف از انجام پروژه

سیستم یه خط تولید محصولات لبنی برای مثال که شامل یه نوار نقاله با کنترل موتور dc و دو تا سنسور یکی سنسور مجاورتی برای فهمیدن رسیدن ظرف خالی به زیر شیر پر کننده محتوای لبنی و سنسور بعدی نیز سنسور تشخیص پر شدن ظروف تا سطح مورد انتظار می باشد.

که برای کاهش هزینه ها و راحتی تست الگوریتم هامون فعلا تصمیم گرفتم که بجای هر کدوم از این سنسور ها یه کلید از اختصاص دهم.

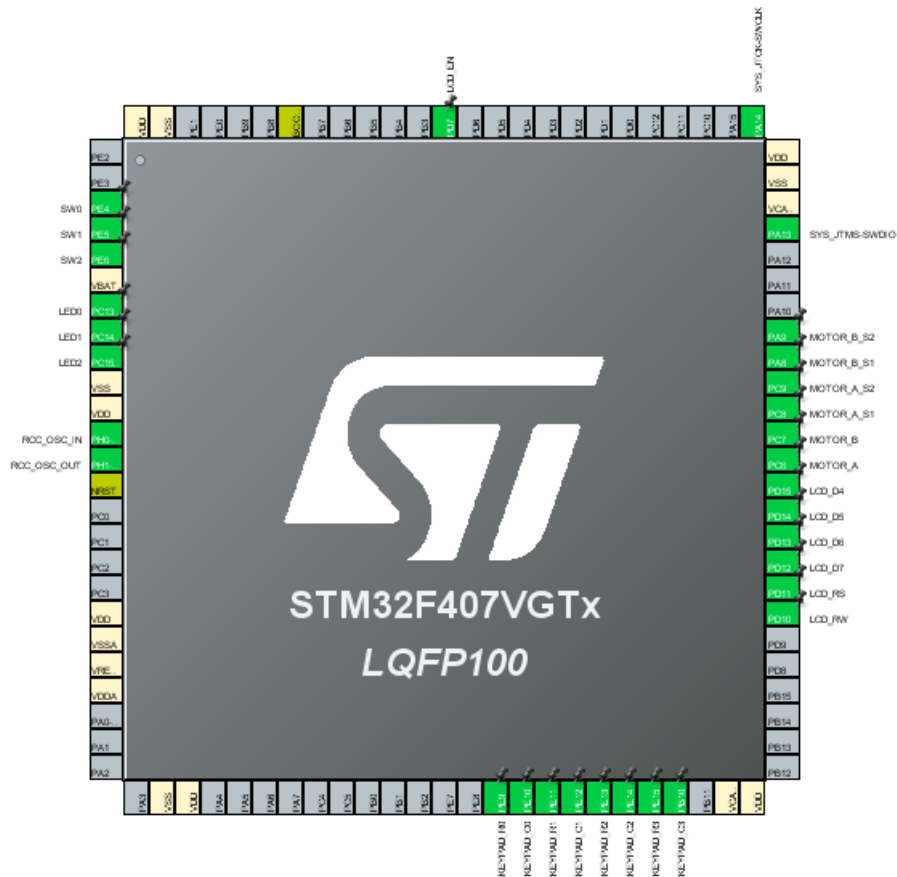
و یه موتور پمپ برای پر کردن ظروف.

### انتخاب پلتفرم برای انجام این پروژه

با توجه به نیاز های این پروژه که حتما می خواستم از یه سیستم عامل برای این پروژه استفاده کنم که بعدا بتونه فیچر های پردازشی رو هم به صورت ریل تایم هندل کنه به سراق میکرو های سری stm32 رفتم.

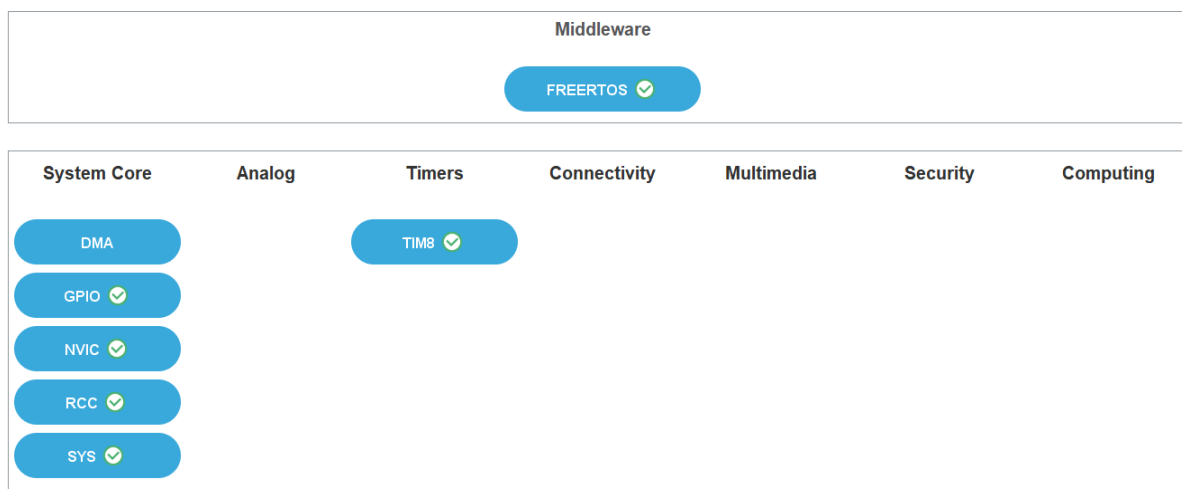
از اونجا که از قبل برد آموزشی stm32f407vgt6 از شرکت نیرا سیستم رو داشتم پس تصمیم گرفتم برای جلو گیری از هزینه بیشتر از همین برد استفاده کنم. که مشخصاتش رو میشه از روی سایت شرکت دید.

برای پلتفرم نرم افزاری هم اومدم از برنامه cubemx برای تولید شاکله پروژه و راه اندازی پریفرال های مورد نیاز استفاده کردم. برای محیط توسعه اصلی هم اومدم از keil ide استفاده کردم. که چون برای خود شرکت ARM هست و رایگانه تقریبا بهترین گزینه من بود.



بخش های عملکرد سیستم:

برای توضیح فرایند اصلی برنامه می خوام از شکل زیر کمک بگیرم:

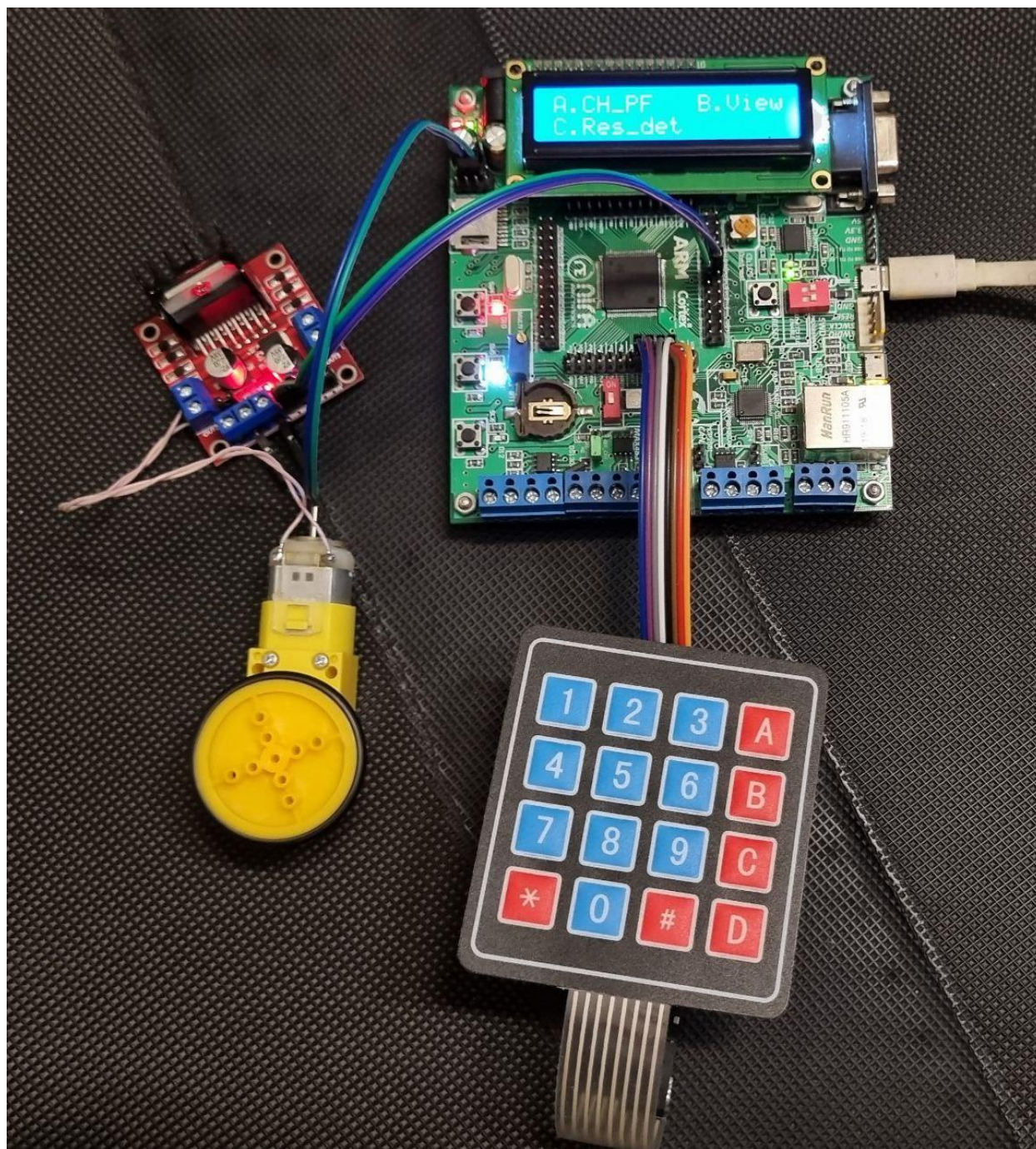


همین طور که از این شکل مشخص است در این پروژه از سیستم عامل freertos استفاده شده است. که از اهداف پروژه هم همین بود.

و پریفرال های مهم بعدی هم اینه که از PWM timer و همین طور واحد GPIO استفاده کردم.

از سخت افزار هایی که استفاده کردم میشه به صفحه کلید ۴ در ۴، نمایشگر کاراکتری ۲ در ۱۶ و یه مازول درایور موتور L298N دارای دو کانال خروجی برای این کار اشاره کرد.

یه تصویر هم از نمای اصلی پروژه در اندازه های دانشجویی خدمتون ارائه میدم. لازم به ذکر است که فایل ویدیویی توضیح سیستم در فایل پروژه قرار دارد و می توانید به اون هم مراجعه کنید.



با استفاده از کلید های A تا D میشه پروژه رو مدیریت کرد.

**کتاب خونه های مورد استفاده**

برای این پروژه از کتاب خونه های زیر استفاده کردم

- Keypad
- Lcd
- Motor\_dc
- Freertos
- HAL

برای انجام پروژه، من از چند تا تسک استفاده کردم:

- تسک دیفالت
- تسک کیپد
- تسک مدیریت سنسورها

در ادامه می خوام به صورت خط به خط به توضیحی از برنامه ارائه بدم:

ابتدا در تابع main برنامه کرنل رو فعال سازی کردم

```

87
88  /* Initialize all configured peripherals */
89  MX_GPIO_Init();
90  MX_TIM8_Init();
91  /* USER CODE BEGIN 2 */
92
93  /* USER CODE END 2 */
94
95  /* Init scheduler */
96  osKernelInitialize(); /* Call init function for freertos objects (in freertos.c) */
97  MX_FREERTOS_Init();
98
99  /* Start scheduler */
100  osKernelStart();
101
102  /* We should never get here as control is now taken by the scheduler */
103  /* Infinite loop */
104  /* USER CODE BEGIN WHILE */
105  while (1)
106  {

```

سپس در تسک دیفالت که از ابتدا برنامه شروع به کار کرده بود می آیین تسک کیپد رو راه اندازی می کنیم و کارهای تسک دیفالت رو ادامه می دهیم، مثل شروع به کار lcd و سنسور هامون



Core > Src > freertos.c > MX\_FREERTOS\_Init(void)

```
123 void MX_FREERTOS_Init(void) {  
148     /* Create the thread(s) */  
149     /* creation of DesplayTask */  
150     DesplayTaskHandle = osThreadNew(DesplayDefaultTask, NULL, &DesplayTask_attributes);  
151  
152
```

```
153     /* USER CODE BEGIN DesplayDefaultTask */  
170 void DesplayDefaultTask(void *argument)  
171 {  
172     /* USER CODE BEGIN DesplayDefaultTask */  
173  
174     menu_MessageQueue_t RecQueMes = {0};  
175  
176  
177     LCD_Init();  
178     osDelay(2);  
179     menu_init();  
180     menu_page_init(MENU_MAIN_PAGE);  
181  
182     /* creation of KeypadTask */  
183     KeypadTaskHandle = osThreadNew(KeypadDefaultTask, NULL, &KeypadTask_attributes);  
184  
185     /* Infinite loop */  
186     while(1)  
187     {  
188  
189         osMessageQueueGet(lcdQueueHandle, &RecQueMes, 0, osWaitForever);  
190         // LastQueMes = &RecQueMes;  
191         menu_process(&RecQueMes);  
192  
193         //osDelay(100);  
194     }  
195     /* USER CODE END DesplayDefaultTask */  
196 }
```

و در اینجا بود که ما برنامه منو رو هم راه اندازی کردیم.

در حلقه درست این تسک هم همیشه منتظر اینیم که یه چیزی در صف قرار داده شده باشه تا اون متوا رو با استفاده از مراتبی که در منو در نظر است رو صفحه نمایش دهد.



```

62 void menu_process (menu_MessageQueue_t * RecQuseMes)
63 {
64     char lcd_buffer[8] = {0};
65
66     switch(RecQuseMes->rq_status)
67     {
68         case CH_PRODUCT_NUM:
69             > switch(g_menu_state) ...
84
85             break;
86         case CH_CURRENT_PF:
87             > switch(g_menu_state) ...
102
103             break;
104         case CH_ENTERED_PF:
105             > switch(g_menu_state) ...
120
121             break;
122         default:
123             break;
124     }
125
126 }

```

حالا به توابع تسک کیپد میرویم:

```

207 /* USER CODE END Header_KeypadDefaultTask */
208 void KeypadDefaultTask(void *argument)
209 {
210     /* USER CODE BEGIN KeypadDefaultTask */
211     keypad_init(FALLING_EDGE);
212
213     /* creation of SensorTask */
214     SensorTaskHandle = osThreadNew(SensorDefaultTask, NULL, &SensorTask_attributes);
215
216     osDelay(100);
217
218     /* Infinite loop */
219     while(1)
220     {
221         keypad_process();
222
223         osDelay(100);
224     }
225     /* USER CODE END KeypadDefaultTask */
226 }
227

```

در ابتدای این تسک ما تابع کیپد رو فعال سازی کردیم که در مواقع رخ داد لبه پایین رونده حساسیت از خود نشان دهد.

و در ادامه تسک سنسور ها رو فعال سازی می کنیم تا پس از شروع به کار کم کم باز پردازی روی پردازنده اعمال شود.

بخش حلقه گیپد هم میاد به بررسی و اسکن صفحه کلید می پردازد.

بزارین کمی هم درباره فرآیند تسک سنسور صحبت کنیم:

```
267
268     if(HAL_GPIO_ReadPin(SW0_GPIO_Port, SW0_Pin) == GPIO_PIN_RESET) //por shod sensor :)
269     {
270         TrsQueMes.rq_status = CH_PRODUCT_NUM;
271         TrsQueMes.product_num++;
272         osMessageQueuePut(lcdQueueHandle, &TrsQueMes, NULL, 50);
273
274
275         motor_dc_set_status(MOTOR_B, DIR_STOP);
276         osDelay(200);
277         motor_dc_set_status(MOTOR_A, DIR_RIGHT);
278
279     }
280     else if(HAL_GPIO_ReadPin(SW1_GPIO_Port, Sw1_Pin) == GPIO_PIN_RESET) //resid sensor :)
281     {
282         motor_dc_set_status(MOTOR_A, DIR_STOP);
283         osDelay(200);
284         motor_dc_set_status(MOTOR_B, DIR_RIGHT);
285
286     }
```

سنسور ها رو برای راحتی کار در پیاده سازی الگوریتم ها اومدیم و به صورت کلید هندل کردیم که اگر هر کدوم از سنسور ها تحریک شود عمل کرد موتور ها نصب به اون تنظیم شود که یا باید موتور نوار نقاله روشن باشد یا باید موتور پمپ روشن باشد که در این تابع کنترل می شود.

برای پیاده سازی کلید های کیپد نسبت به منو هایی که توشون قرار داریم از فرآیند های زیر استفاده میشه که با استفاده از switch case دارد کنترل می شود

```

22
23     uint32_t number = 0;
24
25     __weak void keypad_kp_0_9_func (void)
26     {
27         menu_MessageQueue_t TrsQueMes;
28         uint32_t temp;
29
30         switch(g_menu_state)
31         {
32             case MENU_MAIN_PAGE:
33                 break;
34             case MENU_CHANGE_PF:
35                 TrsQueMes.rq_status = CH_ENTERED_PF;
36                 temp = (number * 10) + get_keypad_value();
37                 if(temp <= 100)
38                 {
39                     number = temp;
40                 }
41                 TrsQueMes.entered_PF = number;
42                 osMessageQueuePut(lcdQueueHandle, &TrsQueMes, NULL, 50);
43                 break;
44             case MENU_VIEW_DETALES:
45                 break;
46             case MENU_RESET_PRODUCTS_NUM:
47                 break;
48             default:
49                 break;
50         }
51

```

همون طور که معلومه فقط در منوی که می خواهیم درضد توان موتور رو وارد کنیم نیاز هست که کلید های ۰ تا ۹ کاری رو انجام بدن.

```

54     __weak void keypad_kp_10_func (void) //kp A //MENU_CHANGE_PF
55     {
56         menu_MessageQueue_t TrsQueMes;
57
58         switch(g_menu_state)
59         {
60             case MENU_MAIN_PAGE:
61                 menu_page_init(MENU_CHANGE_PF);
62                 break;
63             case MENU_CHANGE_PF:
64                 TrsQueMes.rq_status = CH_ENTERED_PF;
65                 number /= 10;
66                 TrsQueMes.entered_PF = number;
67                 osMessageQueuePut(lcdQueueHandle, &TrsQueMes, NULL, 50);
68                 break;
69             case MENU_VIEW_DETALES:
70                 break;
71             case MENU_RESET_PRODUCTS_NUM:
72                 break;
73             default:
74                 break;
75         }
76

```

برای کلید شماره A هم نیاز هست که اگر توی منوی اصلی بودیم ما رو به منوی change pf ببرد و اگر توی منوی تغییر توان بودیم از عدد وارد شده بیاد و آخرین عدد رو کسر کنه.

```
79 __weak void keypad_kp_11_func (void) //kp B //MENU_VIEW_DETALES //set MOTOR_A PF
80 {
81     menu_MessageQueue_t TrsQueMes;
82
83     switch(g_menu_state)
84     {
85         case MENU_MAIN_PAGE:
86             menu_page_init(MENU_VIEW_DETALES);
87             break;
88         case MENU_CHANGE_PF:
89             motor_dc_set_pf(MOTOR_A, (uint8_t)number); //set MOTOR_A PF
90             TrsQueMes.rq_status = CH_CURRENT_PF;
91             TrsQueMes.current_PF = motor_dc_get_pf(MOTOR_A);
92             osMessageQueuePut(lcdQueueHandle, &TrsQueMes, NULL, 50);
93             osDelay(1000);
94             menu_page_init(MENU_MAIN_PAGE);
95             break;
96         case MENU_VIEW_DETALES:
97             break;
98         case MENU_RESET_PRODUCTS_NUM:
99             break;
100        default:
101            break;
102    }
103 }
```

کلید شماره B هم میاد و در منوی تغییر توان تنظیمات رو ست می کنه. و در منوی اصلی میاد وارد صفحه گزارش می کنه که دیتیل رو بتونیم ببینیم.

```
106 __weak void keypad_kp_12_func (void) //kp C //MENU_RESET_PRODUCTS_NUM
107 {
108
109     switch(g_menu_state)
110     {
111         case MENU_MAIN_PAGE:
112             menu_page_init(MENU_RESET_PRODUCTS_NUM);
113             break;
114         case MENU_CHANGE_PF:
115             break;
116         case MENU_VIEW_DETALES:
117             break;
118         case MENU_RESET_PRODUCTS_NUM:
119             break;
120         default:
121             break;
122     }
123 }
```

کلید C هم میاد و دیتیل رو ریست می کنه و اینکه بزارین یه توضیحی هم برای این تابع راه اندازی صفحه منو بدم:

```

23 void menu_page_init (menu_state_t menu_state)
24 {
25     char lcd_buffer[17] = {0};
26
27     g_menu_state = menu_state;
28
29     switch(menu_state)
30     {
31         case MENU_MAIN_PAGE:
32             LCD_Clear();
33             LCD_Puts(0, 0, "A.CH_PF");           //ta 6 poreh //kp 10
34             LCD_Puts(10, 0, "B.View");           //ta 13 poreh //kp 11
35             LCD_Puts(0, 1, "C.Res_det");         //ta 10 poreh //kp 12
36             break;
37         case MENU_CHANGE_PF:
38             LCD_Clear();
39             LCD_Puts(0, 0, "Enter new %PF:");
40             number = 0;
41             break;
42         case MENU_VIEW_DETALES:
43             LCD_Clear();
44             sprintf(lcd_buffer, "Produc = %d", TrsQueMes.product_num);
45             LCD_Puts(0, 0, lcd_buffer);
46             TrsQueMes.current_PF = motor_dc_get_pf(MOTOR_A);
47             sprintf(lcd_buffer, "PF = %d", TrsQueMes.current_PF);
48             LCD_Puts(0, 1, lcd_buffer);
49             break;
50         case MENU_RESET_PRODUCTS_NUM:
51             need_res_prod_num = 1;
52             LCD_Clear();
53             LCD_Puts(0, 0, "detales reset");
54             LCD_Puts(0, 1, "successfully.");
55             osDelay(1000);
56             menu_page_init(MENU_MAIN_PAGE);
57             break;
58     }
59 }
60

```

می بینیم که با توجه به اینکه قراره به چه صفحه ای وارد شدیم دیتای همون صفحه رو روی نمایش گر نمایش بده.

```
126  __weak void keypad_kp_13_func (void) //kp D //MENU_MAIN_PAGE
127  {
128
129      switch(g_menu_state)
130      {
131          case MENU_MAIN_PAGE:
132              break;
133          case MENU_CHANGE_PF:
134              menu_page_init(MENU_MAIN_PAGE);
135              break;
136          case MENU_VIEW_DETALES:
137              menu_page_init(MENU_MAIN_PAGE);
138              break;
139          case MENU_RESET_PRODUCTS_NUM:
140              menu_page_init(MENU_MAIN_PAGE);
141              break;
142          default:
143              break;
```

کلید شماره D هم توی همه پیج ها میاد و منو رو به منوی اصلی تغییر می دهد.

خب گزارش ما تا همین جا به پایان رسید. لطفا برای داشتن حس واقعی تر از فرایند سیستم ویدیو موجود در فایل پروژه رو مشاهده کنید.