

마이크로프로세서응용실험 프로젝트 보고서

20171520 김태훈, 20181536 엄석훈

0. Board1 GPIO 설정

```
62 RCC->APB2ENR |= 0x00000015;
63 GPIOA->CRL = 0x40304444;
64 GPIOA->CRH = 0x444B84B4;
65 GPIOA->ODR |= 0x00000800;
66 GPIOC->CRL = 0x33338888;
67 GPIOC->ODR = 0x0000000F;
68 AFIO->MAPR = 0x00004000;
```

그림 1. GPIO 설정

: board1에서 GPIO의 PA0~3를 input floating으로 사용하고, PA5는 board의 LED를 켜거나 끄는 push-pull 출력단자로 사용하기 위한 설정을 하였다. 그림1의 LINE64에서는 PA9는 USART1의 Tx으로 사용하기 위해 AF output으로 설정하고, PA11은 USART1_CTS를 RTS에 대한 입력으로 사용하며, 이때 RTS가 연결이 끊겨도 pull-up으로 사용할 수 있도록 하였다. 다음으로 key matrix의 col에는 PC0~3을 pull-up 입력단자로 사용하고, PC4~7은 해당 row에 대한 버튼이 눌렸는지를 감지하는 push-pull 출력단자로 사용하기 위한 설정을 하였다. 마지막으로 AFIO_MAPR의 0x00004000을 써서, PA11를 USART1_CTS으로 사용하기 위해 CANRX를 사용하지 않는 설정을 하였다.

1. BOARD 1 - ADC

```
70 RCC->APB2ENR |= 0x00000200;
71 ADC1->CR1 = 0x00000020;
72 ADC1->CR2 = 0x001E0002;
73 ADC1->SMPR2 = 0x00007E00;
74 ADC1->SQR1 = 0x00000000;
75 ADC1->SQR2 = 0x00000000;
76 ADC1->SQR3 = 0x00000004;
77 NVIC->ISER[0] |= 0x00040000;
78 ADC1->CR2 |= 0x00000001;
79 ADC1->CR2 |= 0x00400000;
```

그림 2. ADC1의 초기설정

: ADC1을 사용하여 온도센서의 결과와 조도센서의 결과를 연속적으로 ADC된 결과를 EOC ISR을 통해 값을 얻기 위한 설정을 그림1과 같이 설정했다.

```
363 void ADC1_2_IRQHandler(void) {
364     if(ADC1->SR & 0x02) {
365         if(ADC_flag == 0) {
366             ADC1->SQR3 = 0x00000006;
367             ADC_flag = 1;
368             ADCConverted_Temp = ADC1->DR;
369         }
370         else {
371             ADC1->SQR3 = 0x00000004;
372             ADC_flag = 0;
373             ADCConverted_Illum = ADC1->DR;
374         }
375     }
376     return;
377 }
```

그림 3. ADC1_2_IRQHandler 코드

```
132 void calc_adc(void) {
133     int temp, illum;
134
135     temp = (ADCConverted_Temp * 3600 / 4095 - 500);
136     temp1 = (unsigned char)(temp / 10);
137     temp2 = (unsigned char)(temp % 10);
138     temp3 = (unsigned char)(temp * 10 / 100 + 32);
139     temp4 = (unsigned char)((temp * 10 / 10) % 10);
140
141     if(ADCConverted_Illum <= 3015) {
142         illum = ((ADCConverted_Illum * 13010 / 4095) - 2020);
143         if(illum < 0) {
144             illum = 50;
145         }
146     }
147     else {
148         illum = (ADCConverted_Illum * 44659 / 100 - 1693640);
149     }
150     illum1 = (unsigned char)(illum / 1000);
151     illum2 = (unsigned char)((illum - illum1 * 1000) / 10);
152     illum3 = (unsigned char)(illum % 10);
153     return;
154 }
155 }
```

그림 4. calc_adc 함수 코드

: 그림3에서 PA4에는 온도센서의 출력을 연결하고, PA6에는 조도센서의 출력을 연결하여, ADC1의 IN4, IN6을 통해 ADC1에 analog값이 전달되도록 하였고, conversion이 끝남에 따라 EOC를 통해

다음으로 그림4에서는 온도센서, 조도센서의 ADC값을 각 단위에 맞게 display하기 위한 calibration을 하는 코드이다. 이때 온도는 ADC의 값이 linear한 관계로 구할 수 있고, temp1은 섭씨온도의 십의 자리와 일의 자리, temp2는 섭씨 온도의 소수점 자리를 의미한다. 조도의 경우, ADC 값과 Lux의 관계를 2개의 일차함수 형태로 근사화해서 구했다. Illim1은 Lux의 천의 자리, 백의 자리를 의미하고, illum2는 Lux의 십의 자리, 일의 자리를 의미하며, illum3은 Lux의 소수점 자리를 의미한다.

```
92 RCC->APB1ENR |= 0x00000001;
93 TIM2->CR1 = 0x00000004; //key pad scan
94 TIM2->CR2 = 0x00000000;
95 TIM2->PSC = 0x00001C1F;
96 TIM2->ARR = 0x000000C7;
97 TIM2->DIER = 0x00000001;
98 NVIC->ISER[0] |= 0x10000000;
99 TIM2->CR1 |= 0x00000001;
```

```

402 void TIM2_IRQHandler(void) {
403     unsigned int i, temp_key_col;
404
405     if((TIM2->SR & 0x01) != 0) {
406         GPIOC->BRR = (unsigned int)((~(key_row < 4) & 0xF0) | (key_row << 20));
407         for(i = 0; i < 10000; i++) {}
408
409         temp_key_col = GPIOC->IDR;
410         temp_key_col = (temp_key_col) & 0x0F;
411         if(key_pressing == 0 || temp_key_col == 0x0F) {
412             key_col = (unsigned char)(temp_key_col & 0x0F);
413
414             current_pressed_key = key_matrix[key_row][key_col];
415
416             if(key_col != 0x00) {
417                 key_pressing = 1;
418             }
419             else {
420                 key_pressing = 0;
421             }
422
423             if(key_pressing == 0) {
424                 key_row = (unsigned char)(key_row << 1);
425                 if(key_row == 0x10) {
426                     key_row = 0x01;
427                 }
428             }
429
430             TIM2->SR = ~0x00000001;
431         }
432     }
433     return;
434 }

```

```
101 RCC->APB1ENR |= 0x00000002;
102 TIM3->CR1 = 0x00000004; //1초 증가
103 TIM3->CR2 = 0x00000000;
104 TIM3->PSC = 0x00001C1F;
105 TIM3->ARR = 0x0000270F;
106 TIM3->DIER = 0x00000001;
107 NVIC->ISER[0] |= 0x20000000;
108 TIM3->CR1 |= 0x00000001;
```

[illegible]

: TIM3는 1초마다 Update event를 발생시키도록 설정하여 시간에 대한 정보를 업데이트 하는 용

도로 사용하였다. 이에 대한 ISR에서는 만약 time set=1을 통해 시간 설정을 완료된 상태인 경우에 초를 나타내는 ss를 1씩 증가시키며, 만약 ss=60이 되면 분을 담당하는 mm을 1 증가시키고, ss=0으로 초기화 한다. 현재 모드가 시간 설정을 다시 하는 경우가 아니면 현재 시간을 업데이트 하지만, 만약 시간을 재설정하는 경우에는 이전 시간 정보를 업데이트 하는 과정을 추가하여, 시간 재설정을 중지하는 경우에 대비하여 이전 시간이 지속적으로 업데이트 되도록 한다.

추가적으로 ADC의 계산 결과를 1초마다 update 하기 위해서 TIM3의 ISR을 수행할 때마다 ADC 결과를 단위에 맞게 변환하는 함수인 clac_adc()를 호출하였다.

4. BOARD 1 - USART

```
81  RCC->APB2ENR |= 0x00004000;
82  USART1->CR1 = 0x00000000;
83  USART1->CR2 = 0x00000000;
84  USART1->CR3 = 0x00000000;
85  USART1->CR3 = 0x00000600;
86  USART1->BRR = 0x00000EA6;
87  NVIC->ISER[1] |= 0x00000020;
88  USART1->CR1 |= 0x00000008;
89  USART1->CR1 |= 0x00002000;
90  USART1->CR1 |= 0x00000080;
```

그림 9. USART1 초기설정

```
120  RCC->APB2ENR |= 0x00000800;
121  TIM1->CR1 = 0x00000004;
122  TIM1->CR2 = 0x00000000;
123  TIM1->PSC = 0x00001C1F;
124  TIM1->ARR = 0x00001387; //0.5초
125  TIM1->DIER = 0x00000001;
126  NVIC->ISER[0] |= 0x02000000;
127  TIM1->CR1 |= 0x00000001;
```

그림 10. TIM1 초기설정

```
379 void USART1_IRQHandler(void) {
380     if(USART1->SR & 0x200) {
381         if(USART1->SR & 0x00000001) {
382             TIM1->CNT = 0;
383             GPIOA->BSRR = 0x20;
384         }
385         else {
386             USART1->SR &= ~0x00000001;
387             TIM1->CNT = 0;
388             GPIOA->BSRR = 0x20;
389         }
390     }
391     if(USART1->SR & 0x00000001) {
392         if(USART1->DR == txdata[tx_index++]) {
393             if(tx_index == 10) {
394                 tx_index = 0;
395             }
396         }
397     }
398 }
399 return;
400 }
```

그림 11. USART1_IRQHandler

```
492 void TIM1_UP_IRQHandler(void) {
493     if(TIM1->SR & 0x01) != 0) {
494         if(USART1->SR & 0x00000001) {
495             USART1->SR &= ~0x00000001;
496             USART1->SR &= ~0x00000001;
497         }
498         else {
499             if(USART1->SR & 0x00000001) {
500                 GPIOA->BSRR = 0x20;
501                 USART1->SR &= ~0x00000001;
502             }
503         }
504     }
505     TIM1->SR &= ~0x00000001;
506 }
507 }
508 }
509 }
510 }
```

그림 12. TIM1_UP_IRQHandler

```
293 void make_txdata(void) {
294     txdata[0] = 0xFF;
295     txdata[1] = current_mode; //mode
296     if(hh > 12 && current_mode == 3) {
297         txdata[2] = hh - 12;
298     }
299     else {
300         txdata[2] = hh;
301     }
302     txdata[3] = mm; //분
303     txdata[4] = ss; //초
304     txdata[5] = illum1; //조도
305     txdata[6] = illum2; //조도
306     txdata[7] = illum3; //조도
307     if(current_mode == 6) { //화씨
308         txdata[8] = temp3; //온도
309         txdata[9] = temp4; //온도
310     }
311     else { //섭씨
312         txdata[8] = temp1; //온도
313         txdata[9] = temp2; //온도
314     }
315 }
316 return;
317 }
```

그림 13. make_txdata 코드

: 먼저 board1을 USART1_Tx로 사용하기 위해 CR1의 UE=1과 TE=TXEIE=1로 설정하고, CTS를 사용하기 위해 CR3의 CTSE=CTSIE=1로 설정하였다. 먼저, USART1_SR의 CTS의 status가 변함에 따라 CTS에 의한 ISR을 수행할 때에는 TIM1의 CNT=0으로 초기화하고 PA5의 ODR를 1로 set한 상태를 유지하여 board의 led를 켜 상태를 유지해서, 이는 통신이 원활하게 진행되고 있음을 알린다. 만약 CTS가 계속 1이거나 0인 상태가 TIM1의 update event가 발생할 때까지 유지가 되면 이는 통신이 원활하게 진행되고 있지 않음으로 인식하여 TIM1_UP ISR에서 PA5를 toggling하여 LED를 깜빡이는 것을 통해 통신이 이뤄지지 않음을 알리도록 하였다. 다음으로 TXE=1이 됨에 따라 Tx의 데이터 전송하는 ISR을 구성하였고, 이때 Rx단에 전송하는 data는 0xFF라는 데이터를 test로 보내고, 현재 mode, 시, 분, 초, 조도의 천의자리&백의자리, 조도의 십의 자리&일의 자리, 조도의 소수점, 섭씨 온도의 십의자리& 일의자리, 섭씨 온도의 소수점, 화씨 온도의 십의자리&일의자리,

화씨 온도의 소수점을 순차적으로 반복해서 보내게 된다.

5. BOARD 1 – FLASH MEMORY

```
319 void flash_read(void) {  
320     memcpy(&current_mode, (unsigned char*) StartAddr, 1);  
321     default_mode = current_mode;  
322     return;  
323 }  
324 }
```

그림 13 flash_read

```
326 void flash_write(void) {  
327     while((FLASH->SR & 0x01) != 0) {}  
328  
329     if((FLASH->CR & 0x80) != 0) {  
330         FLASH->KEYR = 0x45670123;  
331         FLASH->KEYR = 0xCDEF89AB;  
332     }  
333  
334     FLASH->CR |= 0x00000002;  
335     FLASH->AR = StartAddr;  
336     FLASH->CR |= 0x00000040;  
337  
338     while((FLASH->SR & 0x01) != 0) {}  
339     if((FLASH->SR & 0x20) != 0) {  
340         FLASH->SR |= 0x20;  
341     }  
342     FLASH->CR &= (unsigned int)~0x00000002;  
343  
344     while((FLASH->SR & 0x01) != 0) {}  
345     if((FLASH->CR & 0x80) != 0) {  
346         FLASH->KEYR = 0x45670123;  
347         FLASH->KEYR = 0xCDEF89AB;  
348     }  
349  
350     FLASH->CR |= 0x00000001;  
351     while ((FLASH->SR & 0x01) != 0) {}  
352     *((__IO uint16_t*)(StartAddr)) = (uint16_t)current_mode;  
353     while ((FLASH->SR & 0x01) != 0) {}  
354  
355     if ((FLASH->SR & 0x20) != 0) {  
356         FLASH->SR |= 0x20;  
357     }  
358     FLASH->CR &= (unsigned int)~0x00000001;  
359  
360     return;  
361 }
```

그림 14 flash_write

Flash memory 기능을 구현하기 위해서 flash_read 함수와 flash_wirte 함수를 구현하였다. 먼저 flash_read 함수는 board1이 처음 시작할 때 main함수에서 한 번 호출하여 정해진 메모리 공간에서 모드에 대한 정보를 가져와 current_mode와 default_mode로 불러온다. 다음으로 flash_write 함수는 2개의 절차에 따라 데이터를 저장한다. 먼저 flash가 lock 되어 있다면 line 329~332을 통해 unlock을 해준다. 그리고 page erease를 하기 위해 FLASH->CR의 PER을 1로 set하고 내가 사용하는 메모리 공간을 FLASH->AR에 저장하여 flash를 erase해준다. 그리고 FLASH->SR의 EOP를 확인 후 EOP를 0으로 reset하고 PER를 0으로 reset한다. 다음으로 flash에 현재의 모드를 저장하기 위해서 다시 unlock해주고 FLASH->CR의 PG bit를 1로 set하여 wrtie할 준비를 하고 current_mode값을 StartAddr에 저장해주고 다시 EOP를 0으로 reset하고 PG를 0으로 reset하여 저장하는 절차를 끝낸다. 그리고 이러한 flash_write함수는 current mode를 관리하는 manage_key 함수와 동일하게 current_mode를 관리하는 TIM4의 ISR에서 호출하도록 하였다.

6. BOARD 1 - MODE MANAGE

```

157 void manage_key(void) {
158     int flag = 0;
159     if(current_mode == default_mode) {
160         flag = 1;
161     }
162     if(before_pressed_key != current_pressed_key) {
163         if(before_pressed_key == 0) {
164             TIM4->CNT = 0x00000000;
165             switch(current_pressed_key) {
166                 case 0:
167                     if(current_mode == 1) {
168                         current_mode = 2;
169                     }
170                     else {
171                         current_mode = 2;
172                     }
173                     break;
174                 case 1:
175                     current_mode = 4;
176                     break;
177                 case 2:
178                     if(current_mode == 3) {
179                         current_mode = 6;
180                     }
181                     else {
182                         current_mode = 5;
183                     }
184                     break;
185                 case 3:
186                     if(current_mode == 1) {
187                         time_set = 1;
188                         default_mode = 2;
189                         current_mode = 2;
190                     }
191                     default_mode = current_mode;
192                     TIM4->CNT = 0x00000000;
193                     break;
194                 case 4:
195                     break;
196                 case 5:
197                     index = 1;
198                     if(index == 4) {
199                         index = 1;
200                     }
201                     break;
202                 case 6:
203                     break;
204
205                 case 7:
206                     break;
207                 case 8:
208                     if(current_mode == 1) {
209                         TIM4->CR1 |= 0x00000001;
210                         TIM4->CNT = 0;
211                         TIM4->CR1 |= 0x00000001;
212                     }
213                     if(index == 1) {
214                         hh += 1;
215                         if(hh == 24) {
216                             hh = 0;
217                         }
218                     }
219                     else if(index == 2) {
220                         mm += 1;
221                         if(mm == 60) {
222                             mm = 0;
223                         }
224                     }
225                     else if(index == 3) {
226                         ss += 1;
227                         if(ss == 60) {
228                             ss = 0;
229                         }
230                     }
231                     break;
232                 case 9:
233                     break;
234
235                 case 10:
236                     if(current_mode == 1) {
237                         TIM4->CR1 |= 0x00000001;
238                         TIM4->CNT = 0;
239                         TIM4->CR1 |= 0x00000001;
240                     }
241                     if(index == 1) {
242                         hh = 23;
243                     }
244                     else {
245                         hh = 1;
246                     }
247                     else if(index == 2) {
248                         mm = 1;
249                         if(mm == 0) {
250                             mm = 59;
251                         }
252                     }
253                     else if(index == 3) {
254                         ss = 1;
255                         if(ss == 0) {
256                             ss = 59;
257                         }
258                     }
259                     break;
260                 case 11:
261                     break;
262                 case 12:
263                     break;
264                 case 13:
265                     if(index == 1) {
266                         if(index == 0) {
267                             index = 3;
268                         }
269                         break;
270                 case 14:
271                     break;
272                 case 15:
273                     current_mode = 1;
274                     tempmm = mm;
275                     tempss = ss;
276                     index = 1;
277                     TIM4->CR1 |= 0x00000001;
278                     break;
279                 case 16:
280                     break;
281                 case 17:
282                     break;
283             }
284             before_pressed_key = current_pressed_key;
285             flash_write();
286             if((current_mode != default_mode) && (flag == 1)) {
287                 TIM4->CR1 |= 0x00000001;
288             }
289             return;
290         }
291     }
292 }

```

그림 15 manage_key 함수

Keypad에서 누른 버튼을 통해서 모드를 컨트롤 하기 위해서 manage_key 함수를 사용한다. 전체적인 흐름은 다음과 같다. manage_key 함수는 main함수에서 무한루프를 돌며 호출된다. 그리고 만약 눌린 키가 있고 해당 키가 이전에 눌린 키와 다르다면 해당 키에 따라 모드를 바꿔준다. 0번키는 12/24시간 모드를 토글해주고 1번키는 조도 모드, 2번키는 섭씨/화씨 모드를 토글한다. 그리고 3번키는 엔터키로 동작을 수행하고 나서 해당 동작을 확정해준다. 그리고 5번키는 시계모드에서 시->분->초 순서로 설정할 숫자를 이동해주고 13번키는 반대로 초->분->시 순서로 설정할 숫자를 이동해준다. 그리고 8번키와 10번키는 각각 숫자를 하나씩 늘리거나 감소시킨다. 그리고 15번키는 시간설정모드로 들어간다. 그리고 위에서 말했듯이 flash_wirte 함수를 수행한다. 그리고 매 버튼이 눌렸을 때 화면에 표시되는 current_mode와 확정된 모드인 default_mode가 다르다면 5초동안 키가 안 눌렸는지 확인하는 TIM4를 enable한다.

```

476 void TIM4_IRQHandler(void) {
477     if((TIM4->SR & 0x01) != 0) {
478         if(default_mode == 1 || (current_mode == 1 && default_mode == 2) || (current_mode == 1 && default_mode == 3)) {
479             index = 1;
480             hh = tempmm;
481             mm = tempmm;
482             ss = tempss;
483         }
484         current_mode = default_mode;
485         flash_write();
486         TIM4->CR1 |= 0x00000001;
487         TIM4->SR |= 0x00000001;
488     }
489     return;
490 }
491
492 }

```

그림 16 TIM4_IRQHandler

TIM4을 사용하기 위해 기본적으로 버스를 연결해주고 5초에 한번 update event가 발생하여 interrupt가 일어날 수 있도록 하였다. 먼저 TIM4의 인터럽트가 발생하였다는 것은 current_mode와 default_mode가 다르며 5초동안 아무런 버튼도 눌리지 않았다는 뜻이다. 따라서 TIM4의 ISR에서는 먼저 update event가 pending되었는지 확인한다. 그리고 current_mode를 default_mode로 돌아가도록 하였다. 그리고 TIM4를 disable하였다. 그리고 만약 조건문을 통해서 default_mode가 1이었거나 current_mode가 시간설정모드이고

default_mode가 시간모드라면 저장해두었던 시간을 불러온다. 이 기능을 통해서 시간설정모드에서도 5초 이내에 버튼을 누르지 않아 원래 시간모드로 돌아갈 때 내재적으로 시간이 흐를 수 있도록 하였고 시간설정모드에서 시간설정 중 5초이상 버튼을 누르지 않으면 다시 12:00:00으로 돌아가도록 하였다.

7. BOARD 2- DOT MATRIX

```

140 void make_dotmatrix_buffer(void) {
141     int i;
142     NVIC->ICER[0] = 0x72000020;
143     switch(current_mode) {
144         case 1: //시계
145             case 2:
146             case 3:
147                 for(i = 0; i < 5; i++) {
148                     dotmatrix_buffer[i] = (unsigned int)(font[hb / 10][i] << 24);
149                     dotmatrix_buffer[i] |= (unsigned int)(font[hb % 10][i] << 20);
150                     if(i == 1 || i == 3) {
151                         dotmatrix_buffer[i] |= 0x00000000;
152                     }
153                     dotmatrix_buffer[i] |= (unsigned int)(font[mm / 10][i] << 16);
154                     dotmatrix_buffer[i] |= (unsigned int)(font[mm % 10][i] << 12);
155                     if(i == 2 || i == 3) {
156                         dotmatrix_buffer[i] |= 0x00000100;
157                     }
158                     dotmatrix_buffer[i] |= (unsigned int)(font[ss / 10][i] << 8);
159                     dotmatrix_buffer[i] |= (unsigned int)(font[ss % 10][i] << 4);
160                 }
161                 if(hb >= 10) {
162                     col_len = 28;
163                 }
164                 else {
165                     col_len = 24;
166                 }
167                 break;
168             case 4: //시계
169                 for(i = 0; i < 5; i++) {
170                     dotmatrix_buffer[i] = (unsigned int)(font[illum1 / 10][i] << 24);
171                     dotmatrix_buffer[i] |= (unsigned int)(font[illum1 % 10][i] << 20);
172                     dotmatrix_buffer[i] |= (unsigned int)(font[illum2 / 10][i] << 16);
173                     dotmatrix_buffer[i] |= (unsigned int)(font[illum2 % 10][i] << 12);
174                     if(i == 4) {
175                         dotmatrix_buffer[i] |= 0x00001000;
176                     }
177                     dotmatrix_buffer[i] |= (unsigned int)(font[illum3][i] << 8);
178                     dotmatrix_buffer[i] |= (unsigned int)(font[illum4][i] << 4);
179                     dotmatrix_buffer[i] |= (unsigned int)(font[illum5][i] << 0);
180                 }
181                 if(illum1 >= 10) {
182                     col_len = 30;
183                 }
184                 if(illum1 > 0) {
185                     col_len = 26;
186                 }
187                 else if(illum2 >= 10) {
188                     col_len = 22;
189                 }
190                 else {
191                     col_len = 18;
192                 }
193                 break;
194             case 5: //시계
195                 for(i = 0; i < 5; i++) {
196                     dotmatrix_buffer[i] = (unsigned int)(font[templ / 100][i] << 22);
197                     dotmatrix_buffer[i] |= (unsigned int)(font[templ / 10 % 10][i] << 18);
198                     dotmatrix_buffer[i] |= (unsigned int)(font[templ % 10][i] << 14);
199                     if(i == 4) {
200                         dotmatrix_buffer[i] |= 0x00001000;
201                     }
202                     dotmatrix_buffer[i] |= (unsigned int)(font[temp2][i] << 8);
203                     if(i == 0) {
204                         dotmatrix_buffer[i] |= 0x00000020;
205                     }
206                     else if(i == 1) {
207                         dotmatrix_buffer[i] |= 0x00000080;
208                     }
209                     else if(i == 2) {
210                         dotmatrix_buffer[i] |= 0x00000020;
211                     }
212                     dotmatrix_buffer[i] |= (unsigned int)(font[temp3][i] << 0);
213                 }
214                 if(templ >= 100) {
215                     col_len = 24;
216                 }
217                 else if(templ >= 10) {
218                     col_len = 22;
219                 }
220                 else {
221                     col_len = 18;
222                 }
223                 break;
224             case 6: //시계
225                 for(i = 0; i < 5; i++) {
226                     dotmatrix_buffer[i] = (unsigned int)(font[templ / 100][i] << 22);
227                     dotmatrix_buffer[i] |= (unsigned int)(font[templ / 10 % 10][i] << 18);
228                     dotmatrix_buffer[i] |= (unsigned int)(font[templ % 10][i] << 14);
229                     if(i == 4) {
230                         dotmatrix_buffer[i] |= 0x00001000;
231                     }
232                     dotmatrix_buffer[i] |= (unsigned int)(font[temp2][i] << 8);
233                     if(i == 0) {
234                         dotmatrix_buffer[i] |= 0x00000020;
235                     }
236                     else if(i == 1) {
237                         dotmatrix_buffer[i] |= 0x00000080;
238                     }
239                     else if(i == 2) {
240                         dotmatrix_buffer[i] |= 0x00000020;
241                     }
242                     dotmatrix_buffer[i] |= (unsigned int)(font[temp3][i] << 0);
243                 }
244                 if(templ >= 100) {
245                     col_len = 24;
246                 }
247                 else if(templ >= 10) {
248                     col_len = 22;
249                 }
250                 else {
251                     col_len = 18;
252                 }
253                 break;
254             case 7: //시계
255                 NVIC->ICER[0] = 0x72000020;
256                 switch(current_mode) {
257                     case 1: //시계
258                     case 2:
259                     case 3:
260                     case 4:
261                     case 5:
262                     case 6:
263                     case 7:
264                     case 8:
265                     case 9:
266                     case 10:
267                     case 11:
268                     case 12:
269                     case 13:
270                     case 14:
271                     case 15:
272                     case 16:
273                     case 17:
274                     case 18:
275                     case 19:
276                     case 20:
277                     case 21:
278                     case 22:
279                     case 23:
280                     case 24:
281                     case 25:
282                     case 26:
283                     case 27:
284                     case 28:
285                     case 29:
286                     case 30:
287                     case 31:
288                     case 32:
289                     case 33:
290                     case 34:
291                     case 35:
292                     case 36:
293                     case 37:
294                     case 38:
295                     case 39:
296                     case 40:
297                     case 41:
298                     case 42:
299                     case 43:
300                     case 44:
301                     case 45:
302                     case 46:
303                     case 47:
304                     case 48:
305                     case 49:
306                     case 50:
307                     case 51:
308                     case 52:
309                     case 53:
310                     case 54:
311                     case 55:
312                     case 56:
313                     case 57:
314                     case 58:
315                     case 59:
316                     case 60:
317                     case 61:
318                     case 62:
319                     case 63:
320                     case 64:
321                     case 65:
322                     case 66:
323                     case 67:
324                     case 68:
325                     case 69:
326                     case 70:
327                     case 71:
328                     case 72:
329                     case 73:
330                     case 74:
331                     case 75:
332                     case 76:
333                     case 77:
334                     case 78:
335                     case 79:
336                     case 80:
337                     case 81:
338                     case 82:
339                     case 83:
340                     case 84:
341                     case 85:
342                     case 86:
343                     case 87:
344                     case 88:
345                     case 89:
346                     case 90:
347                     case 91:
348                     case 92:
349                     case 93:
350                     case 94:
351                     case 95:
352                     case 96:
353                     case 97:
354                     case 98:
355                     case 99:
356                     case 100:
357                     case 101:
358                     case 102:
359                     case 103:
360                     case 104:
361                     case 105:
362                     case 106:
363                     case 107:
364                     case 108:
365                     case 109:
366                     case 110:
367                     case 111:
368                     case 112:
369                     case 113:
370                     case 114:
371                     case 115:
372                     case 116:
373                     case 117:
374                     case 118:
375                     case 119:
376                     case 120:
377                     case 121:
378                     case 122:
379                     case 123:
380                     case 124:
381                     case 125:
382                     case 126:
383                     case 127:
384                     case 128:
385                     case 129:
386                     case 130:
387                     case 131:
388                     case 132:
389                     case 133:
390                     case 134:
391                     case 135:
392                     case 136:
393                     case 137:
394                     case 138:
395                     case 139:
396                     case 140:
397                     case 141:
398                     case 142:
399                     case 143:
400                     case 144:
401                     case 145:
402                     case 146:
403                     case 147:
404                     case 148:
405                     case 149:
406                     case 150:
407                     case 151:
408                     case 152:
409                     case 153:
410                     case 154:
411                     case 155:
412                     case 156:
413                     case 157:
414                     case 158:
415                     case 159:
416                     case 160:
417                     case 161:
418                     case 162:
419                     case 163:
420                     case 164:
421                     case 165:
422                     case 166:
423                     case 167:
424                     case 168:
425                     case 169:
426                     case 170:
427                     case 171:
428                     case 172:
429                     case 173:
430                     case 174:
431                     case 175:
432                     case 176:
433                     case 177:
434                     case 178:
435                     case 179:
436                     case 180:
437                     case 181:
438                     case 182:
439                     case 183:
440                     case 184:
441                     case 185:
442                     case 186:
443                     case 187:
444                     case 188:
445                     case 189:
446                     case 190:
447                     case 191:
448                     case 192:
449                     case 193:
450                     case 194:
451                     case 195:
452                     case 196:
453                     case 197:
454                     case 198:
455                     case 199:
456                     case 200:
457                     case 201:
458                     case 202:
459                     case 203:
460                     case 204:
461                     case 205:
462                     case 206:
463                     case 207:
464                     case 208:
465                     case 209:
466                     case 210:
467                     case 211:
468                     case 212:
469                     case 213:
470                     case 214:
471                     case 215:
472                     case 216:
473                     case 217:
474                     case 218:
475                     case 219:
476                     case 220:
477                     case 221:
478                     case 222:
479                     case 223:
480                     case 224:
481                     case 225:
482                     case 226:
483                     case 227:
484                     case 228:
485                     case 229:
486                     case 230:
487                     case 231:
488                     case 232:
489                     case 233:
490                     case 234:
491                     case 235:
492                     case 236:
493                     case 237:
494                     case 238:
495                     case 239:
496                     case 240:
497                     case 241:
498                     case 242:
499                     case 243:
500                     case 244:
501                     case 245:
502                     case 246:
503                     case 247:
504                     case 248:
505                     case 249:
506                     case 250:
507                     case 251:
508                     case 252:
509                     case 253:
510                     case 254:
511                     case 255:
512                     case 256:
513                     case 257:
514                     case 258:
515                     case 259:
516                     case 260:
517                     case 261:
518                     case 262:
519                     case 263:
520                     case 264:
521                     case 265:
522                     case 266:
523                     case 267:
524                     case 268:
525                     case 269:
526                     case 270:
527                     case 271:
528                     case 272:
529                     case 273:
530                     case 274:
531                     case 275:
532                     case 276:
533                     case 277:
534                     case 278:
535                     case 279:
536                     case 280:
537                     case 281:
538                     case 282:
539                     case 283:
540                     case 284:
541                     case 285:
542                     case 286:
543                     case 287:
544                     case 288:
545                     case 289:
546                     case 290:
547                     case 291:
548                     case 292:
549                     case 293:
550                     case 294:
551                     case 295:
552                     case 296:
553                     case 297:
554                     case 298:
555                     case 299:
556                     case 300:
557                     case 301:
558                     case 302:
559                     case 303:
560                     case 304:
561                     case 305:
562                     case 306:
563                     case 307:
564                     case 308:
565                     case 309:
566                     case 310:
567                     case 311:
568                     case 312:
569                     case 313:
570                     case 314:
571                     case 315:
572                     case 316:
573                     case 317:
574                     case 318:
575                     case 319:
576                     case 320:
577                     case 321:
578                     case 322:
579                     case 323:
580                     case 324:
581                     case 325:
582                     case 326:
583                     case 327:
584                     case 328:
585                     case 329:
586                     case 330:
587                     case 331:
588                     case 332:
589                     case 333:
590                     case 334:
591                     case 335:
592                     case 336:
593                     case 337:
594                     case 338:
595                     case 339:
596                     case 340:
597                     case 341:
598                     case 342:
599                     case 343:
600                     case 344:
601                     case 345:
602                     case 346:
603                     case 347:
604                     case 348:
605                     case 349:
606                     case 350:
607                     case 351:
608                     case 352:
609                     case 353:
610                     case 354:
611                     case 355:
612                     case 356:
613                     case 357:
614                     case 358:
615                     case 359:
616                     case 360:
617                     case 361:
618                     case 362:
619                     case 363:
620                     case 364:
621                     case 365:
622                     case 366:
623                     case 367:
624                     case 368:
625                     case 369:
626                     case 370:
627                     case 371:
628                     case 372:
629                     case 373:
630                     case 374:
631                     case 375:
632                     case 376:
633                     case 377:
634                     case 378:
635                     case 379:
636                     case 380:
637                     case 381:
638                     case 382:
639                     case 383:
640                     case 384:
641                     case 385:
642                     case 386:
643                     case 387:
644                     case 388:
645                     case 389:
646                     case 390:
647                     case 391:
648                     case 392:
649                     case 393:
650                     case 394:
651                     case 395:
652                     case 396:
653                     case 397:
654                     case 398:
655                     case 399:
656                     case 400:
657                     case 401:
658                     case 402:
659                     case 403:
660                     case 404:
661                     case 405:
662                     case 406:
663                     case 407:
664                     case 408:
665                     case 409:
666                     case 410:
667                     case 411:
668                     case 412:
669                     case 413:
670                     case 414:
671                     case 415:
672                     case 416:
673                     case 417:
674                     case 418:
675                     case 419:
676                     case 420:
677                     case 421:
678                     case 422:
679                     case 423:
680                     case 424:
681                     case 425:
682                     case 426:
683                     case 427:
684                     case 428:
685                     case 429:
686                     case 430:
687                     case 431:
688                     case 432:
689                     case 433:
690                     case 434:
691                     case 435:
692                     case 436:
693                     case 437:
694                     case 438:
695                     case 439:
696                     case 440:
697                     case 441:
698                     case 442:
699                     case 443:
700                     case 444:
701                     case 445:
702                     case 446:
703                     case 447:
704                     case 448:
705                     case 449:
706                     case 450:
707                     case 451:
708                     case 452:
709                     case 453:
710                     case 454:
711                     case 455:
712                     case 456:
713                     case 457:
714                     case 458:
715                     case 459:
716                     case 460:
717                     case 461:
718                     case 462:
719                     case 463:
720                     case 464:
721                     case 465:
722                     case 466:
723                     case 467:
724                     case 468:
725                     case 469:
726                     case 470:
727                     case 471:
728                     case 472:
729                     case 473:
730                     case 474:
731                     case 475:
732                     case 476:
733                     case 477:
734                     case 478:
735                     case 479:
736                     case 480:
737                     case 481:
738                     case 482:
739                     case 483:
740                     case 484:
741                     case 485:
742                     case 486:
743                     case 487:
744                     case 488:
745                     case 489:
746                     case 490:
747                     case 491:
748                     case 492:
749                     case 493:
750                     case 494:
751                     case 495:
752                     case 496:
753                     case 497:
754                     case 498:
755                     case 499:
756                     case 500:
757                     case 501:
758                     case 502:
759                     case 503:
760                     case 504:
761                     case 505:
762                     case 506:
763                     case 507:
764                     case 508:
765                     case 509:
766                     case 510:
767                     case 511:
768                     case 512:
769                     case 513:
770                     case 514:
771                     case 515:
772                     case 516:
773                     case 517:
774                     case 518:
775                     case 519:
776                     case 520:
777                     case 521:
778                     case 522:
779                     case 523:
780                     case 524:
781                     case 525:
782                     case 526:
783                     case 527:
784                     case 528:
785                     case 529:
786                     case 530:
787                     case 531:
788                     case 532:
789                     case 533:
790                     case 534:
791                     case 535:
792                     case 536:
793                     case 537:
794                     case 538:
795                     case 539:
796                     case 540:
797                     case 541:
798                     case 542:
799                     case 543:
800                     case 544:
801                     case 545:
802                     case 546:
803                     case 547:
804                     case 548:
805                     case 549:
806                     case 550:
807                     case 551:
808                     case 552:
809                     case 553:
810                     case 554:
811                     case 555:
812                     case 556:
813                     case 557:
814                     case 558:
815                     case 559:
816                     case 560:
817                     case 561:
818                     case 562:
819                     case 563:
820                     case 564:
821                     case 565:
822                     case 566:
823                     case 567:
824                     case 568:
825                     case 569:
826                     case 570:
827                     case 571:
828                     case 572:
829                     case 573:
830                     case 574:
831                     case 575:
832                     case 576:
833                     case 577:
834                     case 578:
835                     case 579:
836                     case 580:
837                     case 581:
838                     case 582:
839                     case 583:
840                     case 584:
841                     case 585:
842                     case 586:
843                     case 587:
844                     case 588:
845                     case 589:
846                     case 590:
847                     case 591:
848                     case 592:
849                     case 593:
850                     case 594:
851                     case 595:
852                     case 596:
853                     case 597:
854                     case 598:
855                     case 599:
856                     case 600:
857                     case 601:
858                     case 602:
859                     case 603:
860                     case 604:
861                     case 605:
862                     case 606:
863                     case 607:
864                     case 608:
865                     case 609:
866                     case 610:
867                     case 611:
868                     case 612:
869                     case 613:
870                     case 614:
871                     case 615:
872                     case 616:
873                     case 617:
874                     case 618:
875                     case 619:
876                     case 620:
877                     case 621:
878                     case 622:
879                     case 623:
880                     case 624:
881                     case 625:
882                     case 626:
883                     case 627:
884                     case 628:
885                     case 629:
886                     case 630:
887                     case 631:
888                     case 632:
889                     case 633:
890                     case 634:
891                     case 635:
892                     case 636:
893                     case 637:
894                     case 638:
895                     case 639:
896                     case 640:
897                     case 641:
898                     case 642:
899                     case 643:
900                     case 644:
901                     case 645:
902                     case 646:
903                     case 647:
904                     case 648:
905                     case 649:
906                     case 650:
907                     case 651:
908                     case 652:
909                     case 653:
910                     case 654:
911                     case 655:
912                     case 656:
913                     case 657:
914                     case 658:
915                     case 659:
916                     case 660:
917                     case 661:
918                     case 662:
919                     case 663:
920                     case 664:
921                     case 665:
922                     case 666:
923                     case 667:
924                     case 668:
925                     case 669:
926                     case 670:
927                     case 671:
928                     case 672:
929                     case 673:
930                     case 674:
931                     case 675:
932                     case 676:
933                     case 677:
934                     case 678:
935                     case 679:
936                     case 680:
937                     case 681:
938                     case 682:
939                     case 683:
940                     case 684:
941                     case 685:
942                     case 686:
943                     case 687:
944                     case 688:
945                     case 689:
946                     case 690:
947                     case 691:
948                     case 692:
949                     case 693:
950                     case 694:
951                     case 695:
952                     case 696:
953                     case 697:
954                     case 698:
955                     case 699:
956                     case 700:
957                     case 701:
958                     case 702:
959                     case 703:
960                     case 704:
961                     case 705:
962                     case 706:
963                     case 707:
964                     case 708:
965                     case 709:
966                     case 710:
967                     case 711:
968                     case 712:
969                     case 713:
970                     case 714:
971                     case 715:
972                     case 716:
973                     case 717:
974                     case 718:
975                     case 719:
976                     case 720:
977                     case 721:
978                     case 722:
979                     case 723:
980                     case 724:
981                     case 725:
982                     case 726:
983                     case 727:
984                     case 728:
985                     case 729:
986                     case 730:
987                     case 731:
988                     case 732:
989                     case 733:
990                     case 734:
991                     case 735:
992                     case 736:
993                     case 737:
994                     case 738:
995                     case 739:
996                     case 740:
997                     case 741:
998                     case 742:
999                     case 743:
1000                    case 744:

```

그림 17 make_dotmatrix_buffer

Dot matrix를 컨트롤 하기 위해서 make_dotmatrix_buffer함수를 사용하였다. 해당 함수는 현재 모드에 따라서 동작하는데 코드가 길어 보이지만 각 모드에 따라서 dotmatrix에서 출력해야 하는 데이터의 구성을 dotmatrix_buffer에 저장한다. 또한 해당 dotmatrix_buffer의 길이를 col_len에 저장한다.

```

292 void TIM2_IRQHandler(void) {
293     if((TIM2->SR & 0x01) != 0) {
294         GPIOB->ODR = (unsigned int)print_buffer[1][row2] << 8;
295         GPIOC->ODR = (unsigned int)print_buffer[0][row2];
296
297         if(row2 == 5 && led_flag == 0) {
298             GPIOC->ODR |= (0xFF << 8);
299         }
300         else {
301             GPIOC->ODR |= (unsigned int)(-row << 8);
302         }
303
304         row = (unsigned char)(row << 1);
305         if(row == 0x40) {
306             row = 0x01;
307         }
308         row2 += 1;
309         if(row2 == 6) {
310             row2 = 0;
311         }
312
313         TIM2->SR &= ~0x00000001;
314     }
315     else if((TIM2->SR & 0x02) != 0) {
316         GPIOC->ODR |= (0xFF << 8);
317         TIM2->SR &= ~0x00000002;
318     }
319     return;
320 }
321
322
323 void TIM3_IRQHandler(void) {
324     if((TIM3->SR & 0x01) != 0) {
325         if(current_mode == 1 || (hh == 12 && mm == 0 && ss == 0)) == 0) {
326             led_flag = 1;
327             TIM3->SR &= ~0x00000001;
328             return;
329         }
330         if(led_flag == 1) {
331             led_flag = 0;
332         }
333         else {
334             led_flag = 1;
335         }
336         TIM3->SR &= ~0x00000001;
337     }
338     return;
339 }
340
341
342 void TIM4_IRQHandler(void) {
343     int i;
344     unsigned int temp_buffer;
345
346     if((TIM4->SR & 0x01) != 0) {
347         col_index = col_len - 16;
348
349         for(i = 0; i < 5; i++) {
350             temp_buffer = (dotmatrix_buffer[i] << scroll_index) >> col_index;
351             print_buffer[i][1] = (temp_buffer >> 8) & 0xFF;
352             print_buffer[i][2] = temp_buffer & 0xFF;
353         }
354         print_buffer[i][5] = (unsigned char)(i << (6 - current_mode));
355         scroll_index++;
356         if(scroll_index > col_len) {
357             scroll_index = 0;
358         }
359         TIM4->SR &= ~0x00000001;
360     }
361     return;
362 }
363
364

```

그림 18 TIM2_IRQHandler

그림 19 TIM3_IRQHandler

그림 20 TIM4_IRQHandler

또한 dot matrix를 컨트롤 하기 위해서 TIM2, TIM3, TIM4를 사용한다. 먼저 TIM2는 dot matrix의 row를 scan해주는 함수이다. 실제로 dot matrix에 표시해야 할 print_buffer를 PC, PB에 저장하여 출력한다. 그리고 해당 데이터를 바탕으로 row를 매번 TIM2 인터럽트가 발생할 때 마다 row를 증가시켜준다. 다음으로 TIM3는 시간설정이 되지 않았을 때 dot matrix의 상태표시줄을 점멸시켜주는 동작을 한다. 간단하게 아직 시간설정이 되지 않은 상태라면 led_flag를 토글하면서 상태표시줄을 점멸시켜준다. 마지막으로 TIM4는 dot matrix에 나타날 숫자를 스크롤링 해주는 함수이다. scroll_index를 증가시키면서 실제 dot matrix에 표시할 print_buffer에 현재 데이터의 dot matrix정보인 dotmatrix_buffer를 scroll_index값에 따라서 복사해준다.

8. BOARD 2 – USART

```

115 void get_data(void) {
116     if(rxdata[0] == 0xFF) {
117         if(current_mode != rxdata[1]) {
118             current_mode = rxdata[1];
119             scroll_index = 0;
120         }
121         hh = rxdata[2];
122         mm = rxdata[3];
123         ss = rxdata[4];
124         illum1 = rxdata[5];
125         illum2 = rxdata[6];
126         illum3 = rxdata[7];
127         temp1 = rxdata[8];
128         temp2 = rxdata[9];
129         illum = illum1 * 100 + illum2;
130         illum = 700 + illum * 10;
131         if(illum > 7000) {
132             illum = 7000;
133         }
134         TIM2->CCR1 = (unsigned short)illum;
135     }
136     return;
137 }
138

```

그림 10 get_data

```

261 void USART1_IRQHandler(void) {
262     unsigned Char temp;
263
264     if(USART1->SR & 0x20) {
265         temp = (unsigned char)USART1->DR;
266         GPIOA->BSRR = 0x10000000;
267         if(rx_index == 0 && temp == 255) {
268             rxdata[rx_index++] = temp;
269         }
270         else if(rx_index > 0) {
271             rxdata[rx_index++] = temp;
272             if(rx_index == 10) {
273                 rx_index = 0;
274             }
275         }
276         if(usart_off == 0) {
277             TIM1->CNT = 0;
278             GPIOA->BSRR = 0x20;
279         }
280         else {
281             usart_off = 0;
282             TIM1->CNT = 0;
283             GPIOA->BSRR = 0x20;
284             TIM2->CCR1 |= 0x00000001;
285         }
286     }
287     return;
288 }
289

```

그림 11 USART1_IRQHandler

```

366 void TIM1_UP_IRQHandler(void) {
367     if((TIM1->SR & 0x01) != 0) {
368         if(usart_off == 0) {
369             usart_off = 1;
370             usart_off_led = 0;
371         }
372         else {
373             if(usart_off_led == 0) {
374                 GPIOA->BSRR = 0x2000000;
375                 usart_off_led = 1;
376             }
377             else {
378                 GPIOA->BSRR = 0x20;
379                 usart_off_led = 0;
380             }
381             GPIOC->ODR |= (0xFF << 8);
382             TIM2->CCR1 &= ~0x00000001;
383             TIM1->SR &= ~0x00000001;
384         }
385     }
386 }
387

```

그림 12 TIM1_UP_IRQHandler

마지막으로 board2의 usart를 구현하기 위해서 usart를 통해서 들어온 데이터가 rxdata 배열에 저장되면 해당 데이터를 바탕으로 시간, 조도, 온도 변수에 저장해준다. 그리고 UART1 ISR에서는 들어온 데이터가 있으면 인터럽트가 발생하여 해당 데이터를 temp로 읽어들인다. 이때 temp가 미리 약속한 255면 첫번째 데이터가 수신되었다고 판단하고 들어오는 데이터를 rxdata에 저장하기 시작한다. 그리고 usart가 끊어졌을 때 이를 판단하기 위해서 TIM1을 사용하였다. 0.5초동안 데이터가 수신되지 않으면 TIM1 인터럽트가 발생하여 PA5의 값을 토글하며 LED를 켜다켰다 해준다. 그리고 usart에서 데이터가 들어오면 TIM1->CNT를 0으로 초기화 하고 항상 PA5의 LED를 켜도록 하였다.