

EXTRAPPOINT2:

Application Note

(Specification 3)

Setup dell'ambiente *Keil µVision* e caricamento sulla scheda *LandTiger*

Collegare la scheda LandTiger al computer e assicurarsi che questa sia accesa.

Aprire il progetto *sample* all'interno della cartella *extrapoint2_board* e, una volta compilato, caricare il codice sulla scheda attraverso l'apposita icona di LOAD

Potenziometro: Funzionamento e Utilizzo all'interno del progetto

Il potenziometro è uno dei devices presenti sulla scheda LandTiger ed è connesso all'ADC attraverso il canale AD0.5. Tale dispositivo è in grado di fornire in input al convertitore un valore analogico compreso tra 0 e 3.3V.

Per un corretto funzionamento dell'ADC, viene per prima cosa invocata la funzione *ADC_init()* nel *main* la quale si occupa del setup del convertitore. Essa imposta quindi la funzionalità del relativo pin (P1.31), il valore del clock, ne abilita il power e la capacità di generare interrupt

```

4  /* Function that initializes ADC
5   */
6  void ADC_init (void) {
7
8      /* Function that initializes ADC
9      */
10     LPC_PINCON->PINSEL3 |= (3UL<<30); /* P1.31 is AD0.5
11
12     LPC_SC->PCOMP |= (1<<12); /* Enable power to ADC block
13
14     LPC_ADC->ADCR = (1<< 5) | /* select AD0.5 pin
15         (4<< 3) | /* ADC clock is 25MHz/5
16         (1<<1); /* enable ADC
17
18     LPC_ADC->ADINTEN = (1<< 8); /* global enable interrupt
19
20     NVIC_EnableIRQ(ADC_IRQn); /* enable ADC Interrupt
21
22 }

```

Table 83. Pin function select register 3 (PINSEL3 - address 0x4002 C00C) bit description _continued

PINSEL3	Pin name	Function when 00	Function when 01	Function when 10	Function when 11	Reset value
11:10	P1.21 _{UJ}	GPIO Port 1.21	MCABORT	PWM1.3	SSEL0	00
13:12	P1.22	GPIO Port 1.22	MC0B0	USB_PWRD	MAT1.0	00
15:14	P1.23	GPIO Port 1.23	MC1	PWM1.4	MISO0	00
17:16	P1.24	GPIO Port 1.24	MC12	PWM1.5	MOSI0	00
19:18	P1.25	GPIO Port 1.25	MC0A1	Reserved	MAT1.1	00
21:20	P1.26	GPIO Port 1.26	MC0B1	PWM1.6	CAP0.0	00
23:22	P1.27 _{UJ}	GPIO Port 1.27	CLKOUT	USB_OVRRCR	CAP0.1	00
25:24	P1.28	GPIO Port 1.28	MC0A2	PCAP1.0	MAT0.0	00
27:26	P1.29	GPIO Port 1.29	MC0B2	PCAP1.1	MAT0.1	00
29:28	P1.30	GPIO Port 1.30	Reserved	Vbus	AD0.4	00
31:30	P1.31	GPIO Port 1.31	Reserved	SCK1	AD0.5	00

2.4 Analog Input (ADC)

Adjustable potentiometer VR1 is connected to analog channel P1.31 (AD0.5). JP12 jumper is used to enable the potmeter input. VR1 setting provides input voltages between 0V and 3V to the ADC.

Tabel 3 Jumper setting for Analog Input

Jumper	Description
JP12	JP12 is used to enable the VR1 connection to ADC input (P1.31 AD0.5).

VR1 Connected to AD0.5: Jumper short
No ADC input: Jumper removed

A questo punto, attraverso un meccanismo di polling, viene sfruttata la ciclicità del RIT per catturare il valore del potenziometro ogni 50 ms. A tal proposito, nell'handler di questo timer viene lanciata la conversione digitale del valore corrente attraverso la funzione *ADC_start_conversion()*, settando di fatto a 1 un particolare bit del registro ADCR

```

void ADC_start_conversion (void) {
    LPC_ADC->ADCR |= (1<<24); /* Start A/D Conversion
}

```

Nel momento in cui l'ADC è in grado di fornire un valore digitale al processore, un interrupt viene scatenato e immediatamente gestito dal relativo handler nel file *IRQ_adc.c*. In questo punto del codice, il programma si occupa di controllare se il vecchio valore di volume coincide oppure no con il valore appena ricevuto. Nel caso in cui differisca, la variabile globale *CurrentVolume* viene immediatamente aggiornata con il nuovo valore fornito dal convertitore e normalizzato su una base di valori interi tra 0 e 3 compresi. Tale variabile consente di gestire l'aspetto grafico dello speaker presente in alto sullo schermo, disegnando o cancellando il numero di onde rappresentanti appunto il livello del volume

```

20 unsigned short AD_current;
21 unsigned short AD_last = 0x00; /* Last converted value
22 volatile int CurrentVolume = 0;
23 volatile int LastVolume = 0;
24
25 void ADC_IRQHandler(void) {
26
27     AD_current = ((LPC_ADC->ADCR)>>4) & 0xFFFF; /* Read Conversion Result
28     if(AD_current != AD_last){
29         CurrentVolume = AD_current*3/0xFFFF;
30         if(CurrentVolume == 0)
31             switchLastVolume();
32     }
33 }

```

CurrentVolume è inoltre utile per l'effettiva regolazione del volume del segnale acustico uscente dalla cassa. Il file *IRQ_timer.c* contiene al suo interno il codice che implementa la generazione del suono e la variabile *CurrentVolume* viene utilizzata per modulare l'ampiezza della sinusoide

```

267 void TIMER2_IRQHandler (void)
268 {
269     static int sineticks=0;
270
271     /* DAC management */
272     static int currentValue;
273     currentValue = SinTable[sineticks];
274     currentValue -= 410;
275     currentValue /= 1;
276     currentValue += 410;
277     LPC_DAC->DACR = (currentValue*CurrentVolume)/3 <<6;
278     sineticks++;
279     if(sineticks==45) sineticks=0;
280
281     LPC_TIM2->IR = 1; /* clear interrupt flag */
282
283 }

```

References:

- (02_UM10360_user_manual)
- (04_HY-LandTiger_SCH)

Acquaviva Salvatore
Matr: 316931