

Lập trình nhúng cơ bản

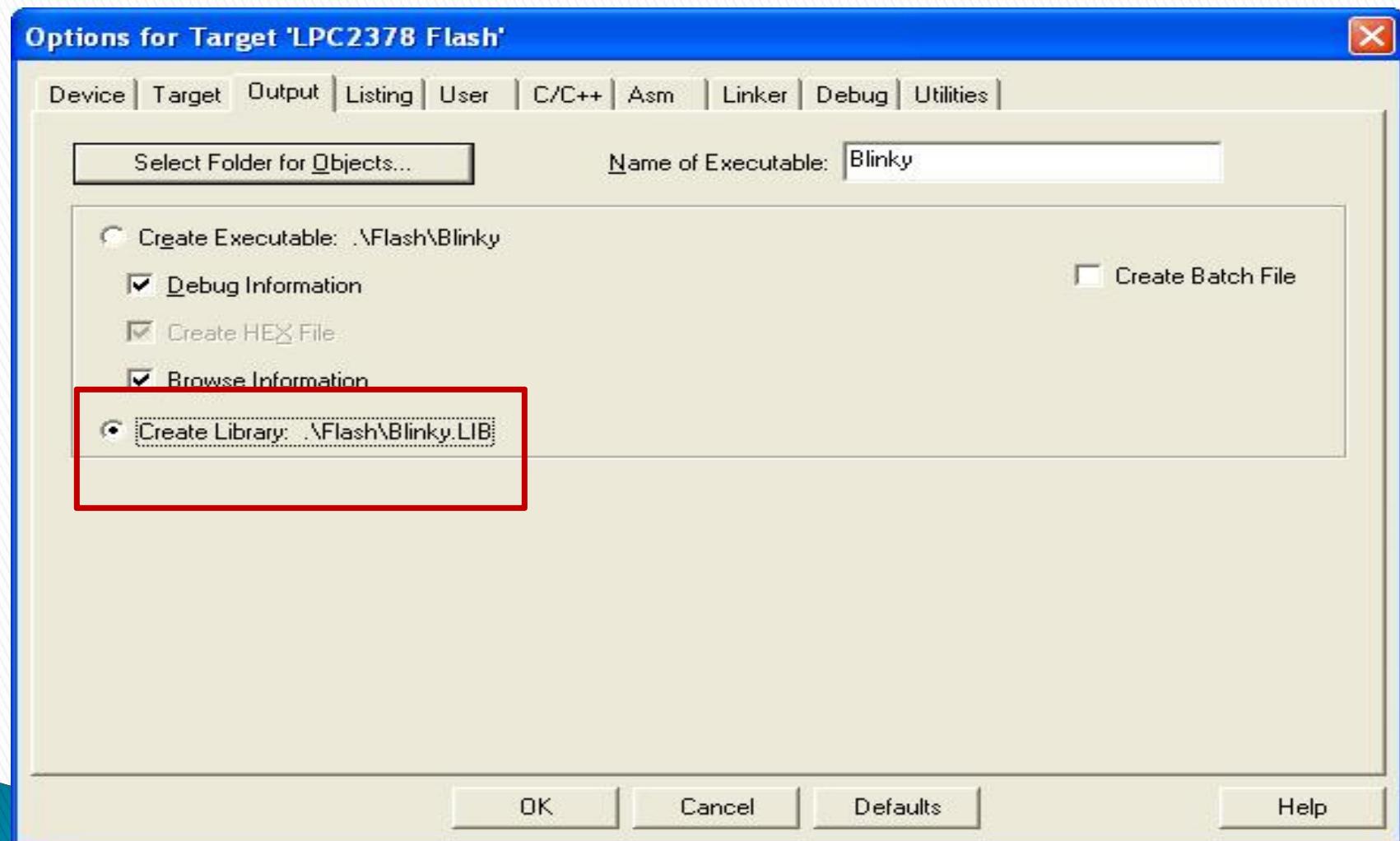
-----Phần thực hành-----

Tuần 7

Nội dung

- ▶ Tạo file *.lib
- ▶ ADC
- ▶ DAC
- ▶ Ví dụ mẫu về ADC, DAC (ứng dụng dùng Speaker)

Tạo file *.lib



Tạo file *.lib

- ▶ Có 1 project, định nghĩa các hàm
- ▶ Biên dịch project này ra file *.lib
- ▶ Một project khác add file .lib này vào và có thể gọi sử dụng các hàm trong file *.lib
- ▶ File *.lib: là file thư viện, giống như DLL lập trình trong Visual Studio

Nội dung

- ▶ Tạo file *.lib
- ▶ ADC
- ▶ DAC
- ▶ Ví dụ mẫu về ADC, DAC

ADC

- ▶ Giới thiệu
- ▶ Các thanh ghi liên quan

Giới thiệu

- ▶ Chuyể
- ▶ LPC2378 hỗ trợ 8 kênh chuyển đổi AD
- ▶ Ví dụ:
 - Đo lượng điện áp của tín hiệu analog đầu vào

Các thanh ghi liên quan

- ▶ AD0OCR (A/D Control Register) : thanh ghi điều khiển AD - chọn chế độ xử lý cho quá trình chuyển đổi AD
- ▶ AD0GDR (A/D Global Data Register): thanh ghi dữ liệu toàn cục - chứa kết quả chuyển đổi AD gần nhất
- ▶ AD0STAT (A/D Status Register): thanh ghi trạng thái - cờ DONE và OVERRUN cho tất cả các kênh

Các thanh ghi liên quan (tt)

- ▶ AD0INTEN (A/D Interrupt Enable Register):
thanh ghi bật ngắt
- ▶ AD0DR n (A/D Channel n Data Register):
thanh ghi chứa dữ liệu tại kênh n – chưa kết
quả chuyển đổi gần nhất trên kênh n

Bật power

Table 46. Power Control for Peripherals register (PCONP - address 0xE01F C0C4) bit description

Bit	Symbol	Description	Reset value
0	-	Unused, always 0.	0
1	PCTIM0	Timer/Counter 0 power/clock control bit.	1
2	PCTIM1	Timer/Counter 1 power/clock control bit.	1
3	PCUART0	UART0 power/clock control bit.	1
4	PCUART1	UART1 power/clock control bit.	1
5	-	Unused, always 0.	1
6	PCPWM1	PWM1 power/clock control bit.	1
7	PCI2C0	The I ² C0 interface power/clock control bit.	1
8	PCSPI	The SPI interface power/clock control bit.	1
9	PCRTC	The RTC power/clock control bit.	1
10	PCSSP1	The SSP1 interface power/clock control bit.	1
11	PCEMC	External Memory Controller	1
12	PCAD	A/D converter (ADC) power/clock control bit.	0

Note: Clear the PDN bit in the AD0CR (see [Section 27–8.1](#)) before clearing this bit, and set this bit before setting PDN.

Cấu hình PINSEL

PINSEL1	Pin name	Function when 00	Function when 01	Function when 10	Function when 11	Reset value
1:0	P0.16	GPIO Port 0.16	RXD1	SSEL0	SSEL	00
3:2	P0.17	GPIO Port 0.17	CTS1	MISO0	MISO	00
5:4	P0.18	GPIO Port 0.18	DCD1	MOSI0	MOSI	00
7:6	P0.19	GPIO Port 0.19	DSR1	MCICLK	SDA1	00
9:8	P0.20	GPIO Port 0.20	DTR1	MCICMD	SCL1	00
11:10	P0.21	GPIO Port 0.21	RI1	MCIPWR	RD1 ^[1]	00
13:12	P0.22	GPIO Port 0.22	RTS1	MCIDAT0	TD1 ^[1]	00
15:14	P0.23	GPIO Port 0.23	AD0.0	I2SRX_CLK	CAP3.0	00
17:16	P0.24	GPIO Port 0.24	AD0.1	I2SRX_WS	CAP3.1	00
19:18	P0.25	GPIO Port 0.25	AD0.2	I2SRX_SDA	TXD3	00
21:20	P0.26	GPIO Port 0.26	AD0.3	AOUT	RXD3	00
23:22	P0.27 ^[2]	GPIO Port 0.27	SDA0	Reserved	Reserved	00
25:24	P0.28 ^[2]	GPIO Port 0.28	SCL0	Reserved	Reserved	00

Đọc giá trị ADC

- ▶ Kết quả chuyển đổi AD được đọc tại thanh ghi AD0DRO (từ bit thứ 6 đến bit thứ 15)

```
value = (AD0DRO >> 6) & 0x3FF;
```

Giá trị điện áp

- ▶ Giá trị điện áp được tính theo công thức

$$\text{Điện áp} = (3,3/\text{ADCValue})/1024$$

DAC

- ▶ Giới thiệu
- ▶ Các thanh ghi liên quan

Giới thiệu

Chuyể
thể hiện ra bên
ngoài (điển hình là âm thanh).

Các thanh ghi liên quan

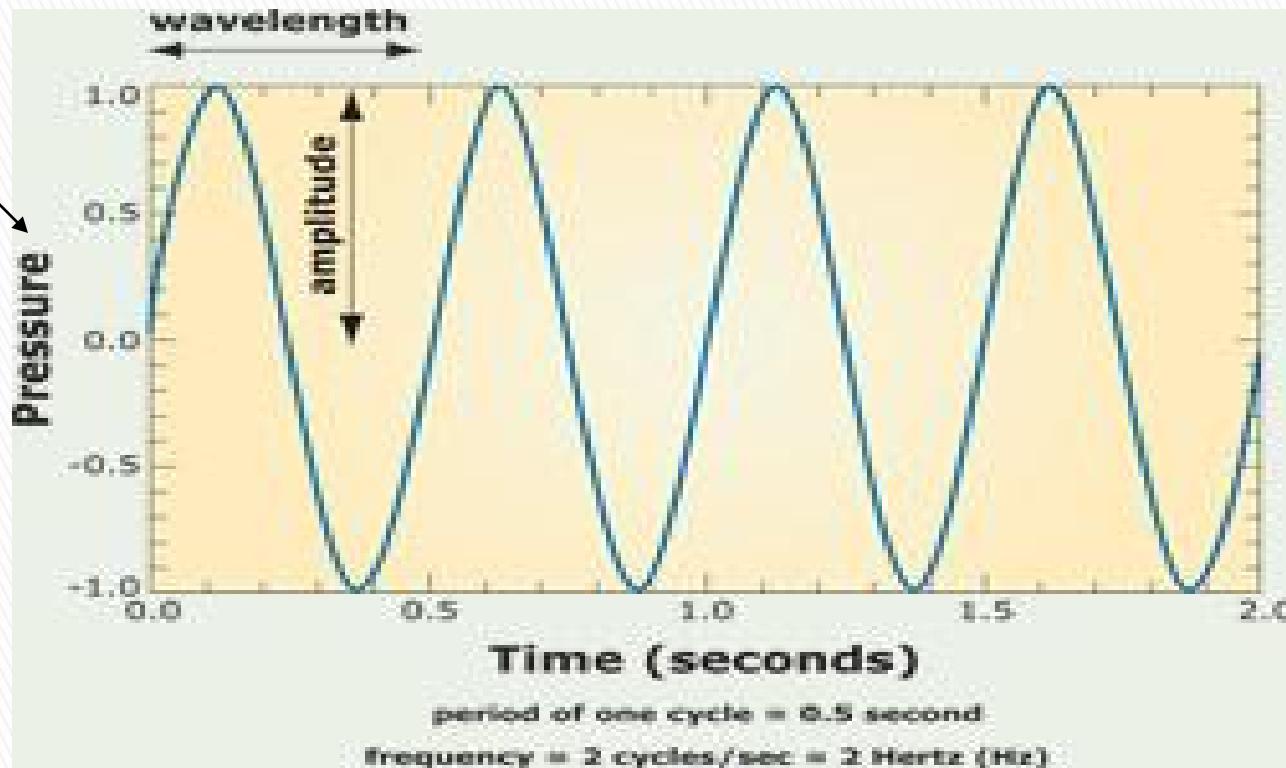
- ▶ DACR (D/A Control Register) : thanh ghi điều khiển AD – để đặt giá trị cần thay đổi tới.

Cấu hình PINSEL

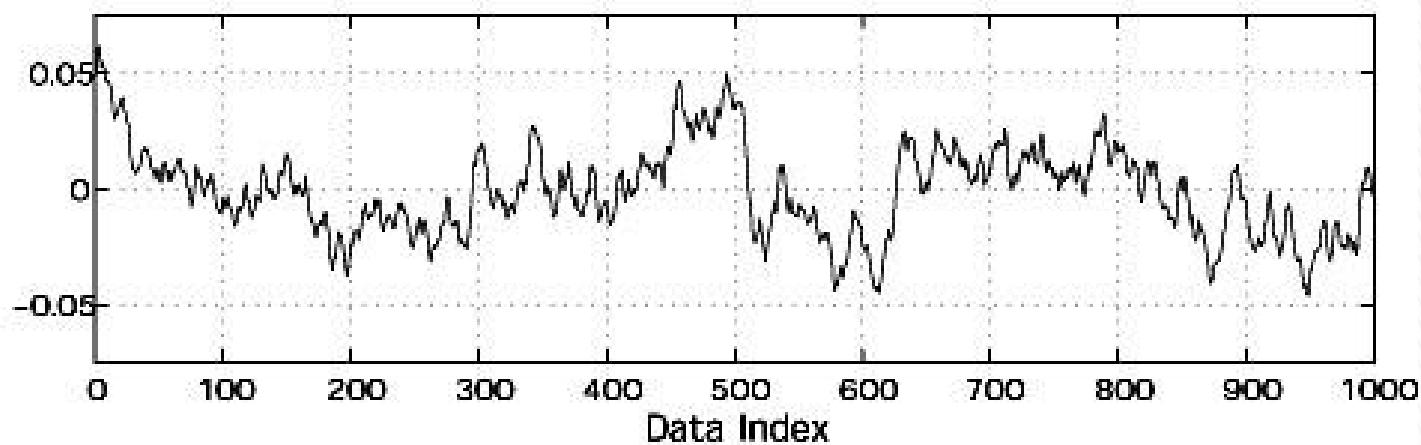
PINSEL1	Pin name	Function when 00	Function when 01	Function when 10	Function when 11	Reset value
1:0	P0.16	GPIO Port 0.16	RXD1	SSEL0	SSEL	00
3:2	P0.17	GPIO Port 0.17	CTS1	MISO0	MISO	00
5:4	P0.18	GPIO Port 0.18	DCD1	MOSI0	MOSI	00
7:6	P0.19	GPIO Port 0.19	DSR1	MCICLK	SDA1	00
9:8	P0.20	GPIO Port 0.20	DTR1	MCICMD	SCL1	00
11:10	P0.21	GPIO Port 0.21	RI1	MCIPWR	RD1 <u>1</u>	00
13:12	P0.22	GPIO Port 0.22	RTS1	MCIDATO	TD1 <u>1</u>	00
15:14	P0.23	GPIO Port 0.23	AD0.0	I2SRX_CLK	CAP3.0	00
17:16	P0.24	GPIO Port 0.24	AD0.1	I2SRX_WS	CAP3.1	00
19:18	P0.25	GPIO Port 0.25	AD0.2	I2SRX_SDA	TXD3	00
21:20	P0.26	GPIO Port 0.26	AD0.3	AOUT	RXD3	00
23:22	P0.27 <u>2</u>	GPIO Port 0.27	SDA0	Reserved	Reserved	00
25:24	P0.28 <u>2</u>	GPIO Port 0.28	SCL0	Reserved	Reserved	00

Mô hình tín hiệu âm thanh

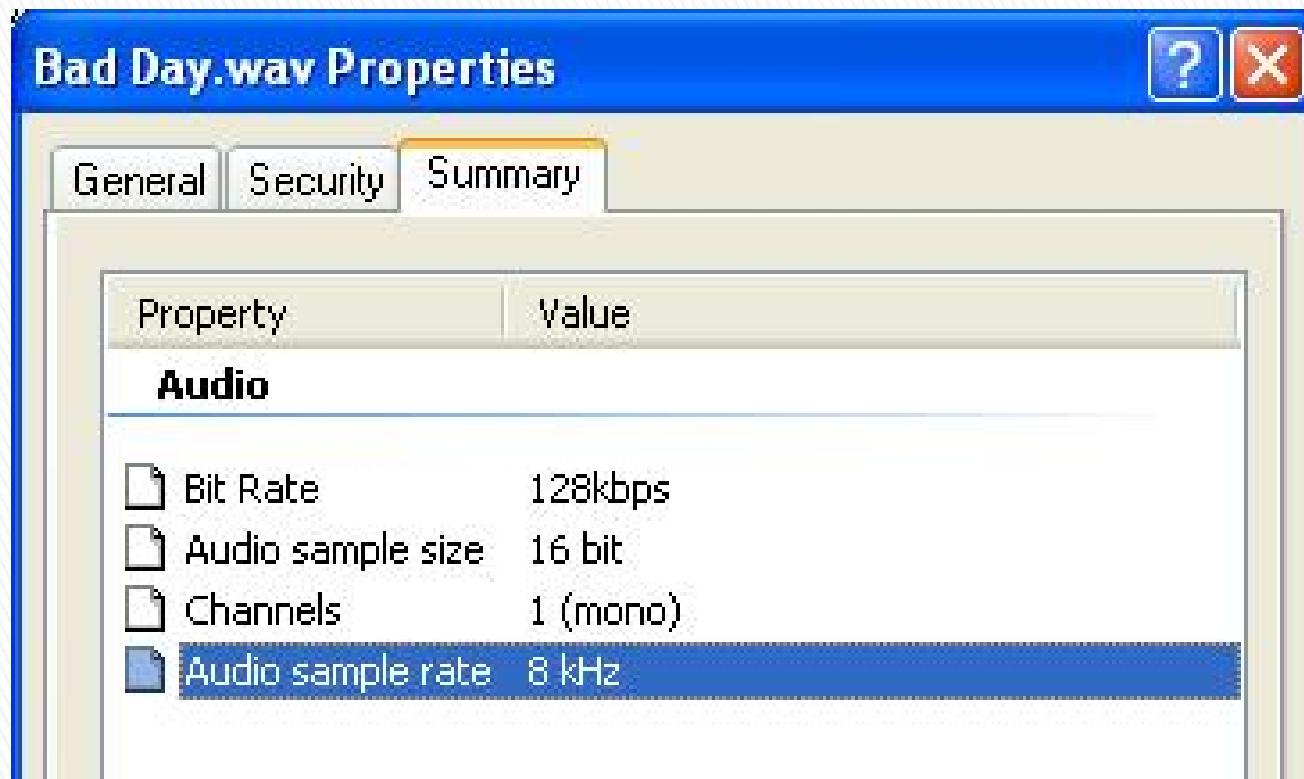
DACR



Mô hình tín hiệu âm thanh



Mô hình tín hiệu âm thanh



Xử lý tín hiệu âm thanh

```
PINSEL1 &= ~(0x03<<20); /* enable DAC */  
PINSEL1 |= (0x02<<20); /* enable DAC */  
  
/* accumulator size / sample rate  
2^32 / 10000 kHz  
429496729 / 10000 = 42950  
#define freq_mult 42950  
efine freq_mult 42950
```

Xử lý tín hiệu âm thanh

```
unsigned short sine_table256[] =
{
0x0200,0x020c,0x0219,0x0225,0x0232,0x023e,0x024b,0x0257,0x0264,0x0270,0x027c,0x0289,0x0295,0x02a1,0x02ad,0x02b8,
0x02c4,0x02d0,0x02db,0x02e7,0x02f2,0x02fd,0x0308,0x0312,0x031d,0x0327,0x0332,0x033c,0x0345,0x034f,0x0358,0x0362,
0x036b,0x0373,0x037c,0x0384,0x038c,0x0394,0x039c,0x03a3,0x03aa,0x03b1,0x03b8,0x03be,0x03c4,0x03ca,0x03cf,0x03d5,
0x03d9,0x03de,0x03e2,0x03e6,0x03ea,0x03ee,0x03f1,0x03f4,0x03f6,0x03f8,0x03fa,0x03fc,0x03fd,0x03fe,0x03ff,0x03ff,
0x03ff,0x03ff,0x03fe,0x03fd,0x03fb,0x03f9,0x03f7,0x03f5,0x03f2,0x03ef,0x03ec,0x03e8,0x03e4,0x03e0,0x03dc,
0x03d7,0x03d2,0x03cd,0x03c7,0x03c1,0x03bb,0x03b4,0x03ae,0x03a7,0x03a0,0x0398,0x0390,0x0388,0x0380,0x0378,0x036f,
0x0366,0x035d,0x0354,0x034a,0x0340,0x0337,0x032c,0x0322,0x0318,0x030d,0x0302,0x02f7,0x02ec,0x02e1,0x02d5,0x02ca,
0x02be,0x02b3,0x02a7,0x029b,0x028f,0x0282,0x0276,0x026a,0x025e,0x0251,0x0245,0x0238,0x022c,0x021f,0x0212,0x0206,
0x01f9,0x01ed,0x01e0,0x01d3,0x01c7,0x01ba,0x01ae,0x01a1,0x0195,0x0189,0x017d,0x0170,0x0164,0x0158,0x014c,0x0141,
0x0135,0x012a,0x011e,0x0113,0x0108,0x00fd,0x00f2,0x00e7,0x00dd,0x00d3,0x00c8,0x00bf,0x00b5,0x00ab,0x00a2,0x0099,
0x0090,0x0087,0x007f,0x0077,0x006f,0x0067,0x005f,0x0058,0x0051,0x004b,0x0044,0x003e,0x0038,0x0032,0x002d,0x0028,
0x0023,0x001f,0x001b,0x0017,0x0013,0x0010,0x000d,0x000a,0x0008,0x0006,0x0004,0x0002,0x0001,0x0000,0x0000,0x0000,
0x0000,0x0000,0x0001,0x0002,0x0003,0x0005,0x0007,0x0009,0x000b,0x000e,0x0011,0x0015,0x0019,0x001d,0x0021,0x0026,
0x002a,0x0030,0x0035,0x003b,0x0041,0x0047,0x004e,0x0055,0x005c,0x0063,0x006b,0x0073,0x007b,0x0083,0x008c,0x0094,
0x009d,0x00a7,0x00b0,0x00ba,0x00c3,0x00cd,0x00d8,0x00e2,0x00ed,0x00f7,0x0102,0x010d,0x0118,0x0124,0x012f,0x013b,
0x0147,0x0152,0x015e,0x016a,0x0176,0x0183,0x018f,0x019b,0x01a8,0x01b4,0x01c1,0x01cd,0x01da,0x01e6,0x01f3,0x0200
};

unsigned long piano_Keys[] =
{
    27,    29,    31,    33,    35,    37,    39,    41,    44,    46,    49,    52,    0,0,0,0,
    55,    58,    62,    65,    69,    73,    78,    82,    87,    92,    98,    104,   0,0,0,0,
   110,   117,   123,   131,   139,   147,   156,   165,   175,   185,   196,   208,   0,0,0,0,
   220,   233,   247,   262,   277,   294,   311,   330,   349,   370,   392,   415,   0,0,0,0,
   440,   466,   494,   523,   554,   587,   622,   659,   698,   740,   784,   831,   0,0,0,0,
   880,   932,   988,  1046,  1109,  1175,  1244,  1318,  1397,  1480,  1568,  1661,  0,0,0,0,
  1760,  1865,  1976,  2093,  2217,  2349,  2489,  2637,  2794,  2960,  3136,  3322,  0,0,0,0,
  3520,  3729,  3951,  4186
};

unsigned char song_lamb[] =
{
0x42,0x40,0x3a,0x40,0x42,0x42,0x40,0x40,0x42,0x45,0x45,
0x42,0x40,0x3a,0x40,0x42,0x42,0x42,0x40,0x40,0x42,0x40,0x3a,
0x42,0x40,0x3a,0x40,0x42,0x42,0x40,0x40,0x42,0x45,0x45.
}
```

Xử lý tín hiệu âm thanh

```
/* phase increment = accumulator size / sample rate * frequency */
accum += freq_mult * freq;

sample = sine_table256[accum>>24]; // 8 bit sine pointer, use for 256 table

/* send the sample to the DAC, shift and mask appropriate bits */
DACR = ((sample) << 6) & 0xFFC0;
```

Bài tập

- ▶ Chơi bản nhạc piano có dữ liệu đã được cung cấp

Chúc các bạn làm bài tốt

