

MSP430開発キット 「ソロバン」

発想から試作へ



ソロバン開発の動機

プロジェクト一覧

- 酸素率測定器(蓄積)
- 紫外線測定器(蓄積)
- 降水量測定(蓄積)
- ワイン畑の温度(蓄積・伝送)



測定器の共通点

- 測定センサ
- ADC
- 処理
- I/O
- 時刻
- 表示
- 通信



ソロバン開発キットの概要

- 開発キットの概念
- ハードウェア
- ソフトウェア
- アプリケーション事例



ソロバン開発キット概念

- 測定器の基本機能を提供する
- センサと処理を分別する
- 小型ボードにより携帯用アプリケーションを可能にする。



ソロバンハードウェア

- MSP430F169/F1611
- RTC (DS 1390)
- SDカード
- 操作ボタン
- グラフィックLCD
- 拡張コネクタ

QuickTime™ and a
decompressor
are needed to see this picture.



特徴一覧

- 2種類 (F169, F1611)
- 周波数8MHz
- 寸法: 80x40
- 重量: 25グラム



ソロバンソフトウェア

- LCDドライバ
- RTCドライバ
- SDカードドライバ
- 体験ライブラリー(FAT16)
- 汎用ソフトウェア



アプリケーション事例

- 酸素率測定器
- 血圧計
- 脳波計
- 心電図(12チャンネル)
- 地震データ蓄積
- 海底ロボット
- 海底油田プラットフォーム安定センサ



無線伝送事例

- 加速時計により傾き測定
- Bluetoothでデータ伝送
- パソコンで受信・表示



ADC12利用

初期化

```
void configure_ADC_interrupts(void) {
    P6SEL = 0x07;           // Inputs A0,A1,A2
    ADC12CTL0 = ADC12ON+MSC+SHT0_2;
    ADC12CTL1 = SHP+CONSEQ_1; // Consecutive
    ADC12MCTL0 = INCH_0;     // X axis
    ADC12MCTL1 = INCH_1;     // Y axis
    ADC12MCTL2 = INCH_2+EOS;
    ADC12IE = 0x04;         // Z as a trigger
    ADC12CTL0 |= ENC;       // Start conversion
    ADC12CTL0 |= ADC12SC;   // Start conversion
    _BIS_SR(LPM0_bits + GIE);
}
```

割り込み関数

```
// ADC12 interrupt service routine
#pragma vector=ADC_VECTOR
__interrupt void ADC12_ISR (void) {
    x = ADC12MEM0;
    y = ADC12MEM1;
    z = ADC12MEM2;
    UpdateGraph(ADC12MEM0);
    MoveTo(0,7);
    WriteString("X accel val: ");
    ClearArea(7, 78, 101);
    MoveTo(78, 7);
    WriteVal(x);
    _BIC_SR_IRQ(LPM0_bits);
}
```



UARTでBluetooth使用

```
void InitUsart0(void) {  
    // P3.4,5 = USART0 TXD/RXD  
    P3SEL |= 0x30;  
    P3DIR |= 0x10; //P3.4 = output  
    P3DIR &= ~0x20; // P3.5 = input  
    // 8-bit, UART  
    U0CTL = (CHAR | SWRST);  
    // UCLK= SMCLK  
    U0TCTL |= SSEL1;  
    // Set baud rate to 9600  
    SetUSART0Rate(9600);  
    // Initialize USART state machine  
    U0CTL &= ~SWRST;  
    // Enable USART0 RX interrupt  
    USART0Enable();  
    Enable_IE_USART0();  
}
```

```
void SendByte(uint8 byte) {  
    IFG1 &= ~UTXIFG0;  
    /* Send the byte */  
    U0TXBUF = byte;  
    /* Wait for the byte to be sent */  
    while (!(IFG1 & UTXIFG0)) { }  
}  
  
uint16 RecvBuffer(uint8 * buffer, uint16 len) {  
    uint16 rcvlen = 0;  
    uint16 t; // Timeout  
    while(rcvlen < len) {  
        timeout = 0;  
        IFG1 &= ~URXIFG0;  
        while(!(IFG1 & URXIFG0) && (t++ < TIME)) { }  
        if(t >= TIME) break;  
        buffer[rcvlen] = U0RXBUF;  
        rcvlen ++;  
    }  
    return rcvlen;  
}
```



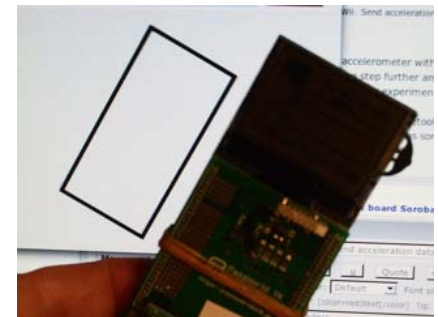
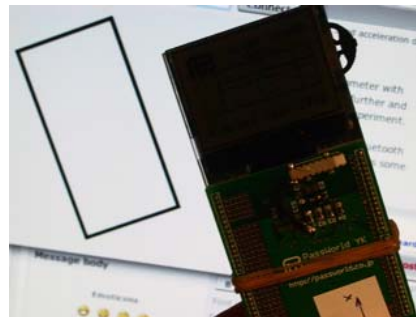
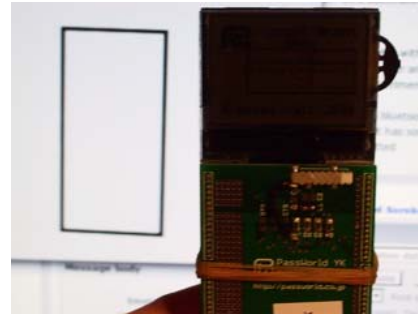
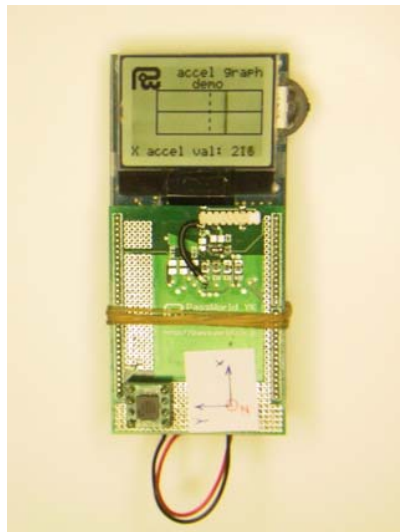
パソコンでデータ受信

動作確認

```
Mercure:~ pascal$ cat /dev/cu.JepicoBluePlum-II-Seria-1
X2790x Y2100y Z2240z
X2793x Y2099y Z2262z
X2799x Y2106y Z2255z
X2756x Y2071y Z2227z
X2793x Y2104y Z2255z
X2797x Y2105y Z2257z
X2776x Y2101y Z2248z
X2805x Y2109y Z2258z
X2805x Y2106y Z2270z
X2785x Y2104y Z2248z
X2812x Y2114y Z2270z
X2796x Y2104y Z2256z
```



結果・デモ



結論

- 携帯可能アプリケーション開発を容易にする
- 1枚で複数試作が可能
- グラフィックにより豊富な表示
- 生産台数が少ない場合に最適

