Module 11

Lecture: Liquid Crystal Display
You will learn in this module

- Busy-wait hardware/software synchronization
- Fundamentals of synchronous serial communication
- How to interface an LCD to TI’s Launchpad Development board
- Software driver (set of functions to create an abstract module)
- Create a minimally intrusive debugging monitor
Hardware/software synchronization

The fundamental problem

- Software executes quickly (48 MHz)
  - Instruction takes 42 ns
- Hardware operates slowly
  - Takes 2 $\mu$s to send 1 byte
  - Takes 14 $\mu$s to output a character

Solutions
- Blind (fixed wait time)
- Busy-wait
- Interrupts (Labs 10, 13, 14)
- Direct memory access
Synchronous Serial Communication on the MSP432

Components
- Enable
- Clock
- Data out
- Data in

MSP432 is master
- Drives clock
- Drives enable
- Initiates transfer

LCD is slave
Synchronous Serial Communication on the MSP432

- Synchronous means send clock and data
  - Send data on one edge of clock
  - Receive data on other edge

- Serial Peripheral Interface (SPI) Protocol
Serial Peripheral Interface (SPI) Timing

Signals

- Clock
- Data out
- Data in
- Enable

n = 7 or 8 bits
Nokia5110 LCD functionality

Monochrome

- Serial Peripheral Interface (SPI)
  - 5 pins
- 84 pixels wide
- 48 pixels high
- 4 MHz speed
- Low cost
LCD Interface

- SPI
  - P9.4 STE
  - P9.5 CLK
  - P9.7 SIMO

- GPIO
  - P9.3 Reset
  - P9.6 Data/command
Decimal output

Output an unsigned integer, n

- Assume n is between 1000 and 9999
- Print as 5 characters, right justified

```c
OutChar(0x20); // space
OutChar(0x30+n/1000); // thousand’s digit
n = n%1000;
OutChar(0x30+n/100); // hundred’s digit
n = n%100;
OutChar(0x30+n/10); // ten’s digit
OutChar(0x30+n%10); // one’s digit
```
Application

LCD provides

1. Debugging information in real time as robot is moving (14 µs/character)
2. Graphical representation of data (optional)

```
Nokia5110_SetCursor(0,2);
Nokia5110_OutString("D= ");
Nokia5110_OutUDec(distance);
Nokia5110_OutString(" mm");
```

```
Nokia5110_SetCursor(3,2);
Nokia5110_OutUDec(distance);
```

Minimally intrusive

\[4+5\times14=74 \mu s\]
Busy-wait synchronization
Synchronous serial communication
Numerical output
Minimally intrusive debugging monitor
Graphics: 4-bit, 16-color BMP
Module 11

Lecture: Organic light-emitting diode display (OLED)
You will learn in this module

- Busy-wait hardware/software synchronization
- Fundamentals of synchronous serial communication
- How to interface an OLED to TI’s Launchpad Development board
- Software driver (set of functions to create an abstract module)
- Create a minimally intrusive debugging monitor
Hardware/software synchronization

The fundamental problem

- Software executes quickly (48 MHz)
  - Instruction takes 42 ns
- Hardware operates slowly
  - Takes 2 µs to send 1 byte
  - Takes 12 µs to output a character

Solutions

- Blind (fixed wait time)
- Busy-wait
- Interrupts (Labs 10, 13, 14)
- Direct memory access
Synchronous Serial Communication on the MSP432

Components
- Enable
- Clock
- Data out
- Data in

MSP432 is master
- Drives clock
- Drives enable
- Initiates transfer

OLED is slave
Synchronous Serial Communication on the MSP432

- Synchronous means send clock and data
  - Send data on one edge of clock
  - Receive data on other edge

- Serial Peripheral Interface (SPI) Protocol

![Diagram of synchronous serial communication on the MSP432](image_url)
Serial Peripheral Interface (SPI) Timing

Signals

- Clock
- Data out
- Data in
- Enable

![Diagram of SPI Timing](image)

n = 7 or 8 bits

CLK
SIMO
SOMI
STE
SSD1306 OLED functionality

Monochrome

- Serial Peripheral Interface (SPI)
  - 5 pins
- 128 pixels wide
- 64 pixels high
- 4 MHz speed
- Low cost
OLED Interface

- **SPI**
  - P9.4 STE
  - P9.5 CLK
  - P9.7 SIMO

- **GPIO**
  - P9.3 Reset
  - P9.6 Data/command
Decimal output

Output an unsigned integer, \( n \)

- Assume \( n \) is between 1000 and 9999
- Print as 5 characters, right justified

```c
OutChar(0x20); // space
OutChar(0x30+n/1000); // thousand’s digit
n = n%1000;
OutChar(0x30+n/100); // hundred’s digit
n = n%100;
OutChar(0x30+n/10); // ten’s digit
OutChar(0x30+n%10); // one’s digit
```
Application

OLED provides

1. Debugging information in real time as robot is moving (12µs/character)

2. Graphical representation of data (optional)

Minimally intrusive

```
SSD1306_SetCursor(0,6);
SSD1306_OutString("th(deg) ");
SSD1306_OutSFix1(theta);
```

```
SSD1306_SetCursor(8,6);
SSD1306_OutSFix1(theta);
```

12+6*12=84 µs
Summary

- Busy-wait synchronization
- Synchronous serial communication
- Numerical output
- Minimally intrusive debugging monitor
- Graphics: 4-bit, 16-color BMP
Module 11

Lecture: UART (for debugging)
Serial Communication

You will learn in this module

- Busy-wait hardware/software synchronization
- Fundamentals of asynchronous serial communication
- Software driver (set of functions to create an abstract module)
- Create a minimally intrusive debugging monitor
Hardware/software synchronization

The fundamental problem

- Software executes quickly (48 MHz)
  - Instruction takes 42 ns
- Hardware operates slowly
  - UART takes 87 µs to output a character

Solutions

- Blind (fixed wait time)
- Busy-wait
- Interrupts (Labs 10, 13, 14)
- Direct memory access
UART Port Selection

MSP432

<table>
<thead>
<tr>
<th>Pin</th>
<th>PxSEL1=0, PxSEL0=1</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1.2</td>
<td>UCA0RXD</td>
</tr>
<tr>
<td>P1.3</td>
<td>UCA0TXD</td>
</tr>
<tr>
<td>P2.2</td>
<td>UCA1RXD</td>
</tr>
<tr>
<td>P2.3</td>
<td>UCA1TXD</td>
</tr>
<tr>
<td>P3.2</td>
<td>UCA2RXD</td>
</tr>
<tr>
<td>P3.3</td>
<td>UCA2TXD</td>
</tr>
<tr>
<td>P9.6</td>
<td>UCA3RXD</td>
</tr>
<tr>
<td>P9.7</td>
<td>UCA3TXD</td>
</tr>
</tbody>
</table>
Universal Asynchronous Receiver/Transmitter (UART)

- **Send/receive a frame**
  - 1 start (low), 5-8 data bits, 1 stop (high)
  - Serial fashion, one bit every **bit-time**
  - No clock is sent, asynchronous, timing derived from data

- **Baud rate** is total number of bits per unit time
  - Baud rate = 1 / bit-time
  - Both transmitter and receiver agree to use the same baud rate

- **Bandwidth** is data or information per unit time
  - Bandwidth = (data-bits / frame-bits) * baud rate
UART - Transmitter

**Software**
- Busy-wait on TXIFG
- Write data to UCA0TXBUF

**Hardware**
- Add start, stop bits
- Shift out at Baud Rate clock
UART - Receiver

**Hardware**
- Wait for start
- Shift in Data at Baud Rate clock
- Check for errors
- Remove start, stop
- Set RXIFG

**Software**
- Busy-wait on RXIFG
- Read data from UCA0RXBUF
Decimal output

Output an unsigned integer, n

- Assume n is between 1000 and 9999
- Print as 5 characters, right justified

```c
OutChar(0x20); // space
OutChar(0x30+n/1000); // thousand’s digit
n = n%1000;
OutChar(0x30+n/100); // hundred’s digit
n = n%100;
OutChar(0x30+n/10); // ten’s digit
OutChar(0x30+n%10); // one’s digit
```
Application

UART serial output provides

1. Debugging information in real time as robot is moving (87 μs/character)
2. Numerical and character information

Moderately intrusive

```
UART_OutString("D= in mm\n");
```

```
4char*87μs/char = 348 μs
```

```
UART_OutUDec(distance);
UART_OutChar('\n');
```

Assume called every 100ms;
Intrusiveness = 348μs/100ms = 0.35%
Summary

- Busy-wait synchronization
- Asynchronous serial communication
- Numerical output
- Moderately intrusive debugging monitor
ti.com/rslk
IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI's products are provided subject to TI's Terms of Sale (www.ti.com/legal/termsofsale.html) or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI's provision of these resources does not expand or otherwise alter TI's applicable warranties or warranty disclaimers for TI products.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2019, Texas Instruments Incorporated