i8255 introduction

Original found on http://home.swipnet.se/~w-16418/

Converted to PDF by HansO, 2001

PPI - Programmable Peripheral Interface

The 8255A programmable peripheral interface (PPI) implements a general-purpose I/O interface to connect peripheral equipment to a microcomputer system bus. The core's functional configuration is programmed by the system software so that external logic is not required to interface peripheral devices.

Features

- Three 8-bit Peripheral Ports - Ports A, B, and C
- Three programming modes for Peripheral Ports: Mode 0 (Basic Input/Output), Mode 1 (Strobed Input/Output), and Mode 2 (Bidirectional)
- Total of 24 programmable I/O lines
- 8-bit bidirectional system data bus with standard microprocessor interface controls

In the SVI-328/318 the 8255 is used to strobe the keyboard lines, to interface joystick fire button switch, the paddle and to control the cassette tape system.

The following I/O ports are used to communicate with the PSG.
### Programming the 8255

The 8255 has 3 8-bit ports (A, B and C), each of which can have a different I/O Status (i.e. input or output). Port C can also be programmed to operate in two halves. (As two separate 4-bit ports). The functional configuration of each port is programmed by the system software. In essence, the CPU "outputs" a control word to the 8255. The control word contains information such as "mode", "bit set", "bit reset", etc, that initializes the functional configuration of the 8255.

There are 3 basic modes of operation under which the ports can function.

- **Mode 0** - Basic Input/Output
- **Mode 1** - Strobed Input/Output (Not used by SVI)
- **Mode 2** - Bi-Directional Bus (Not used by SVI)

In some of these modes port C is used as a control/status port for port A or B. It can be used to confirm when data transfer may take place, and reflect any other flags. The 8255 PPI is therefore supplied with the added option for the user to set or reset any individual bits in port C.

The I/O status, mode of operation and bit setting is defined by the 8255 PPI control byte. (The control byte is accessed using port 97H, 8255 Control port)

The ports may be accessed separately by the CPU.

- Port A is accessed using port 98H
- Port B is accessed using port 99H
- Port C is accessed using port 96H

### NOTE:

**Mode 0** operation:

Mode 0 operation does not require other ports to be used to control data transfer. Data may only be sent to a port if the port is defined as output, and data may only be read from a port when the port is set to input.

### Using the 8255 Ports

**When writing to a port:**

1) Make sure the port is operating as output. (This is done by sending a control byte to
When reading from a port:

1) Make sure the port is operating as input. (This is done by sending a control byte to port 97H).
2) Read the data from the port.
3) Return the port to it's original I/O Status.

**8255 PPI Control Byte**

This byte controls the Input/Output (I/O) status and operating mode of each port.

It also has a secondary function which allows the user to set or reset a particular bit of port C.

To send a control byte to the 8255, simply sent the byte to port 97H. (NOTE: The control byte can only be written, it cannot be read.)

The function to be executed by the 8255 is dependant on the state of bit 7.

If bit 7 is 1, the remaining bits are used to define the I/O status, and operating mode of each port.

If bit 7 is 0, the remaining bits are used to determine what bit is set (set to 1) or reset (set to 0) of port C.

**Configuration control byte**

When bit 7 is set to 1, the command byte operates in the following way:

<table>
<thead>
<tr>
<th>Bit 7: Control byte function (1=Configuration control byte)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bit 6: Operating Mode (Port A and Port C upper)</td>
</tr>
<tr>
<td>Bit 5:</td>
</tr>
<tr>
<td>Bit 4: Port A I/O Status (0: Output, 1: Input)</td>
</tr>
<tr>
<td>Bit 3: Port C (upper) I/O Status (0: Output, 1: Input)</td>
</tr>
<tr>
<td>Bit 2: Operating Mode selection (Port B and Port C lower)</td>
</tr>
<tr>
<td>Bit 1: Port B I/O Status (0: Output, 1: Input)</td>
</tr>
<tr>
<td>Bit 0: Port C (lower) I/O Status (0: Output, 1: Input)</td>
</tr>
</tbody>
</table>

**Port A and Port C (upper) Operating mode**

<table>
<thead>
<tr>
<th>Bit 6</th>
<th>Bit 5</th>
<th>Mode selection</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Mode 0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>Mode 1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Mode 2</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Mode 2</td>
</tr>
</tbody>
</table>
Port B and Port C (lower) operating mode

Bit 2            Mode Selection
-----------------------------
0             Mode 0
1             Mode 1

Bit Set/Reset control byte

When bit 7 is set to 0, the control byte operates in the following way:
Bit 7: Control byte function (0=Bit Set/Reset control byte)
Bit 6: } Not used. (send 0)
Bit 5: }
Bit 4: }
Bit 3: } Bit to change.
Bit 2: }
Bit 1: }
Bit 0: Bit operation
        1: Set bit (make bit=1), 0: Reset bit (make bit=0)

Table to show how bit to change is coded:

<table>
<thead>
<tr>
<th>Bit 3</th>
<th>Bit 2</th>
<th>Bit 1</th>
<th>Bit selected</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Bit 0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>Bit 1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>Bit 2</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>Bit 3</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>Bit 4</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>Bit 5</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>Bit 6</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>Bit 7</td>
</tr>
</tbody>
</table>

Port Usage in SVI-318/328

For each port, the port address (at which the port can be accessed) it's normal I/O
Status (unless changed by the user), it's operating mode (unless changed by the user)
are given.

NOTE: If you are using the firmware, always return the operating modes and I/O
status of the ports used to their status below, otherwise the firmware may crash. (The
firmware expects the operating modes to be the same as given below).

In the SVI-328 is the control word for the 8255 initial set to 10010010

Port A (Port address: 98H) (I/O Status: Input) (Operating mode: 0)
Bit 7: Cassette: Read data
Bit 6: Cassette: Ready
Bit 5: Joystick 2: Trigger
Bit 4: Joystick 1: Trigger
Bit 3: Joystick 2: EOC
Bit 2: Joystick 2: /SENSE
Bit 1: Joystick 1: EOC
Bit 0: Joystick 1: /SENSE
Port B (Port address: 99H) (I/O Status: Input) (Operating mode: 0)
Bit 7: Keyboard: Column status of selected line
Bit 6: Keyboard: Column status of selected line
Bit 5: Keyboard: Column status of selected line
Bit 4: Keyboard: Column status of selected line
Bit 3: Keyboard: Column status of selected line
Bit 2: Keyboard: Column status of selected line
Bit 1: Keyboard: Column status of selected line
Bit 0: Keyboard: Column status of selected line
Port C (Port address: 97H) (I/O Status: Output) (Operating mode: 0)
Bit 7: Keyboard: Click sound bit (pulse)
Bit 6: Cassette: Audio out (pulse)
Bit 5: Cassette: Write data
Bit 4: Cassette: Motor relay control (0=on, 1=off)
Bit 3: Keyboard: Line select 3
Bit 2: Keyboard: Line select 2
Bit 1: Keyboard: Line select 1
Bit 0: Keyboard: Line select 0
Control Port (Port address: 96) (I/O Status: Output) (Operating mode: 0)
Register read (Port address: 9A) (I/O Status: Input) (Operating mode: 0)

Programming Examples

Setting up the 8255
; Sample - Mode select and I/O setup of A,B,C
ppi  EQU  97H
LD  A,92H
OUT (ppi), A
; Sample - Check if CTRL-STOP is pressed
mode  EQU  $9A
portC  EQU  $96
portB  EQU  $99
IN A,(mode) ; Read current PPI mode settings
AND $F0 ; Set bit 15
OR $06
OUT (portC),A ; Write to PPI port C
IN A,(portB) ; Read from PPI port B
AND $22 ; Mask CTRL and STOP keys
RET NZ ; Return, if not pressed

MSX

Usage in MSX: see the MSX Redbook and the MSX(-2) Technical Reference Manual