# Expending Your MCU input/output pins with very cheap and effective solution

## Using Shift Registers

## 74HC165 & 74HC595 Families

| Document Control & History | |
|---|---|
| **Document Title** | **Expending Your MCU input/output pins with very cheap and effective solution Using Shift Registers74HC165 & 74HC595 Families** |
| **Document Ver.** | **1.0 Released on 25-09-2011** |
| **Document Author** | **Razi Raslan, MSc. Biomedical Engineering** |
| **Document Version** | **History** |
| Ver. 1.0 | Initial release correlated to **.mcl** library file version: **1.0.0.0** |
| | |

## Introduction:

The purpose of this library is to control both TTL/CMOS integrated circuits “ICs” 74HC165 and 74HC595 to expand the microcontrollers’’ input/output pins with low cost and effective solution.

## Library Info.:

Status: Release Version: 1.0 Release date: 25/09/2011

Designer: Razi Raslan, MSc. Biomedical Engineering (Razi.Raslan@Gmail.com)



**What is the function of 74HC165?**

It is parallel in, serial out IC which requires only 3 MCU’s pins in order to be controlled so it provide you with 8 inputs. So you will save with this IC 8 – 3 = 5 pins. However, if you cascade more ICs of the same series then you will add 8 inputs times the number of cascaded ICs with no changing with MCU interface. For example, if you use 2 ICs of 74HC165 then you will get 8 x 2 = 16 inputs

**What is the function of 74HC159?**

It is serial in, parallel out ICs which requires only 3 pins from the MCU in order you can control it. It simply let you to expand your controller with 8 additional output pins as a result you will save 8 – 3 = 5 pins. On the other hand, you can cascade as many as you wish of the same IC series with no change with the MCU interface but only expanding the outputs with 8 times the number added of the ICs.

## Library Routines

Before using any library’s routine, then you need to copy the MCL file to the project directory and add the INCLUDE statement in the beginning of your project:

```
Program Testing_New_Library

Include Serial_Port_Using_74xxx
```

Also you need to define the controlling pins connection (interfacing) to the MCU as follows:

```
' For IC: 74HC165
dim SO_165 as sbit sfr external  ' Serial Data Out from the 74HC165 to the MCU
dim PL_LD  as sbit sfr external  ' load data internally from register 1 to 2
dim CP_Clk as sbit sfr external  ' clock

For IC: 74HC595
dim DS_595 as sbit sfr external  ' Serially Data Out From the MCU to 74HC595
dim SH_CP  as sbit sfr external  ' Shift Register Clock (shifting data in)
dim ST_CP  as sbit sfr external  ' Clock to load data from shift regis. to O/P
```

**Example:**

```
' For IC: 74HC165
dim SO_165 as sbit at PortC.0  ' Serial Data Out from the 74HC165 to the MCU
dim PL_LD  as sbit at PortC.1  ' load data internally from register 1 to 2
dim CP_Clk as sbit at PortC.2  ' clock

For IC: 74HC595
dim DS_595 as sbit at PortC.3  ' Serially Data Out From the MCU to 74HC595
dim SH_CP  as sbit at PortC.4  ' Shift Register Clock (shifting data in)
dim ST_CP  as sbit at PortC.5  ' Clock to Load data from shift Register to o/p
```

We still have one line to write before we get ready to use the IC. We need to initialize the IC and how many ICs are cascaded as follows:

```
Initialize_74HC595(1)    ' we are using only one IC of 74HC595
Initialize_74HC165(3)    ' we are using three cascaded ICs of 74HC165
```

Ps. Maximum number of cascaded ICs supported by this library is 4, that is, you can expand the input s up to 4 x8 = 32 inputs and the outputs up to 4 x 8 = 32 outputs!

## How to read the data coming from the 74HC165?

We have four **byte** variables, each variable correspond to one IC of the 74HC165. You do **NOT** need to define them in your code as ***they are already defined for you*** and they are:

Port1_Read , Port2_Read , Port3_Read , Port4_Read

```
Read_Data_From_74HC165()   ' this will force 74HC165 to send its data to MCU

PortB = Port1_Read         ' let us display the data on the LEDs on Port B
```

### How to write (send) the data from the MCU to the 74HC595 Port?

So, after we have initialized the 74HC595 IC, we can use it by sending serially the data to it to display them in parallel mode. In this library you can cascaded up to 4 ICs of the series 74HC595.

```
Port_74HC595_Write_Data(Dim Data1, Data2, Data3, Data4 as byte)
```

The above procedure will send 4 Bytes of data. If you are using only 1 IC of 74HC595, then you need only to fill up the value of Data1 while the remaining 3 bytes you can leave them as zeros.

**Example:**

Let us suppose that we would like to send the value 0x25 so we write

```
Port_74HC595_Write_Data(0x25, 0x00, 0x00, 0x00)  ' send 0x25 to the 74HC595
```