Processing logic signals at the inputs of the microcontroller

Attention Translation Google. If you are experiencing a lack of understanding, please refer to the original.

In modern designs using microcontrollers in the management of the objects have to deal with the processing logic signals, which can contain a large amount of noise. This interference can be caused by contact bounce of mechanical devices, and electromagnetic effects, which may be present in the environment. This is especially true for devices intended for use at industrial sites with high levels of electromagnetic interference.

Classically, to solve similar problems apply the principle of the survey, followed by the expectation of the port and re-examination. But this solution introduces the basic operating cycle of the program a significant delay, which is often not desirable for most projects.

To solve this problem was developed algorithm for processing the input signals, which allows you to filter the signals with duration below a predetermined value, and thus make a minimal delay in the main program loop. This algorithm can be successfully applied not only to control remote sensors signals, but also to survey the buttons control the device.

This algorithm is a signal duration comparator, which lets you change the value of the flag state of the port only when the input signal duration exceeds the set value.

Figure 1 shows an example of such an algorithm. As soon as the duration of the signal exceeds a preset value, the flag port switches.

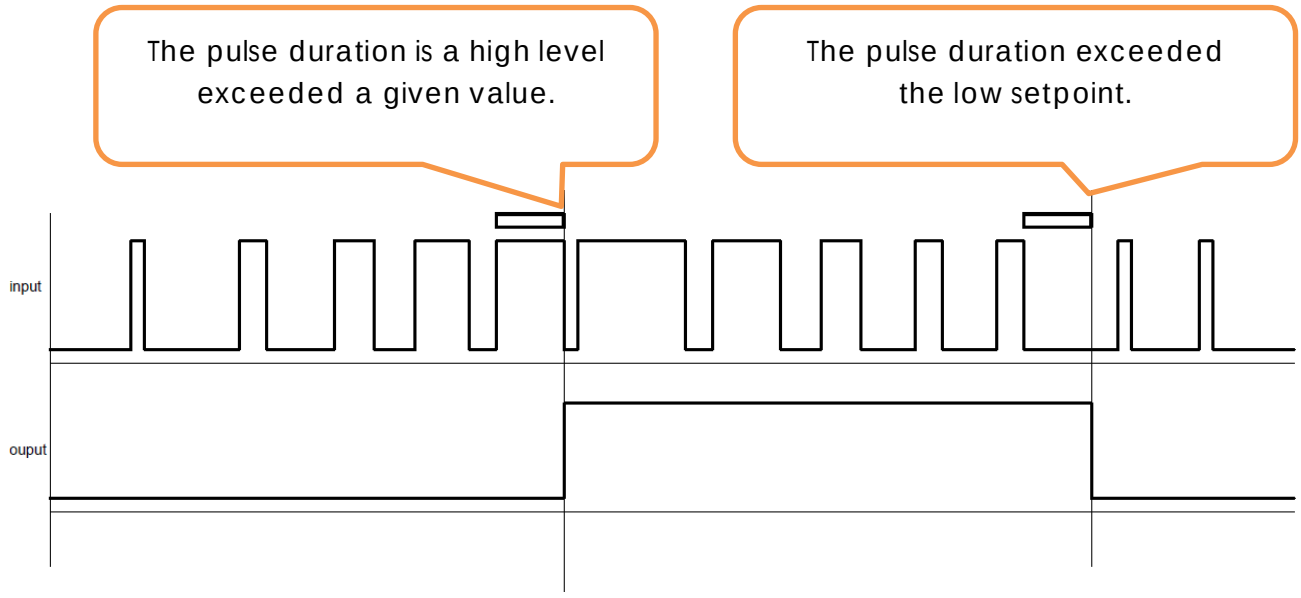


Figure 1

The figure shows that the flag port switch signals whose duration exceeds a predetermined value. You can (if necessary) separately specify the duration of change in state "1" and switch to the duration of the state "0".

To implement this algorithm to allocate a buffer port. This buffer is needed that would allow other features of the program independently using the port, for example, to display data on LED. This buffer is copied to a specific period of the input data. For each line of input bits is allocated and the previous state variable delay timer.

The logic of the next function, it checks the status bit buffer and depending on its current status compares with the state obtained in the previous test. If they do not match, it initializes (loads), the delay timer and sets a bit past the state equal to the current state.

If the current state of the analyzed bit port coincides with the last condition, the function tests the delay timer to zero, and if it is equal to zero, sets the flag.
A similar algorithm and to reset the flag.

In the below given program tested port B. It is understood that included pull-up resistors.

```
if(rPORTB & 0b10000000)        // testing required bit port
{// If the input is zero
   if(bps==1)                  // check the previous state flag
   { // they are not equal
         bps=0;                // zero out the flag
         timerD=ZADINPU;       // initialize the timer delay
   }
   if(timerD ==0)              // check the delay timer == 0
   {// if - Yes
         Flag=1;               // set the flag
   }
}
else
{// if the input high level
   if(bps ==0)                 // testing required bit port
   {// they are not equal
         bps =1;               // set the flag
         timerD =ZADINPU;      // initialize the timer delay
   }
   if(timerD ==0)              // check the delay timer == 0
   {// if - Yes
         Flag =0;              // reset the flag
   }
}
```

Similarly, we can test the keyboard buttons control the device, only that there is no need to test to reset the flag.

```
if(rPORTB & 0b10000000)    // testing required bit port
{       // they are not equal
        if(bps==1)         // check the previous state flag
        { // они не равны
              bps=0;       // zero out the flag
              timerD=ZADINPU;    // initialize the timer delay
        }
        if(timerD ==0)     // check the delay timer == 0
        {       // if - Yes
              Flag=1;      // set the flag
        }
}
else
{       // if the input high level
     bps =1;          // set the flag
     Flag =0;         // reset the flag
```

```
                }
```

Processing delay timer should be placed in the interrupt, which will form the required length. Timer itself is organized as follows.

```
//------------------------ interruptions during 10 Hz.------------------------
                if(timerD!=0) timerD--;
```

To read the port in traps set for reading a block of input signals:

```
//-----------------------------------------------
   // reading the state of the keyboard buttons and sensor inputs
   TRISB=0xFF;                  // Disable output setting to receive data
   _delay(1);                   // delay for normalization of signal levels
                                // to clock frequency higher than 8 MHz, should be increased
   rPORTB=PORTB;                // copying the data port to the buffer
   TRISB=0;                     // setting the port to output
//-----------------------------------------------
```

The logic of copying data to the buffer port may be different and depends on the location of the input port - output.

The demo program that uses this algorithm.

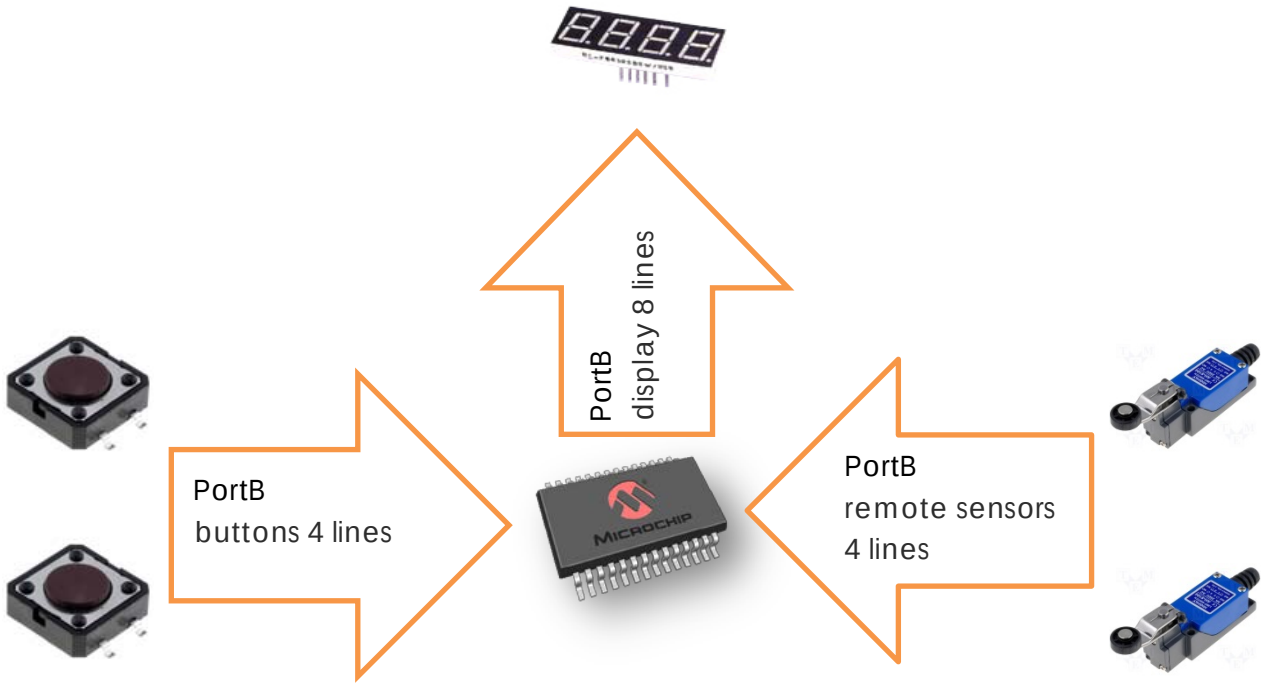


Figure 2

To demonstrate the scheme is used the actual product. In this scheme, PORTB controller is used for dynamic display on the LED display and the same port is used to retrieve information from the four buttons and keyboard with four remote sensors. To handle the bounce clock buttons a delay equal to 0.1 seconds. And for filtering noise from remote sensors duration of 1 second, how to set the flag, and reset the flag.

Controller Clock Speed 8 MHz. Poll PORT B is placed in an interrupt.

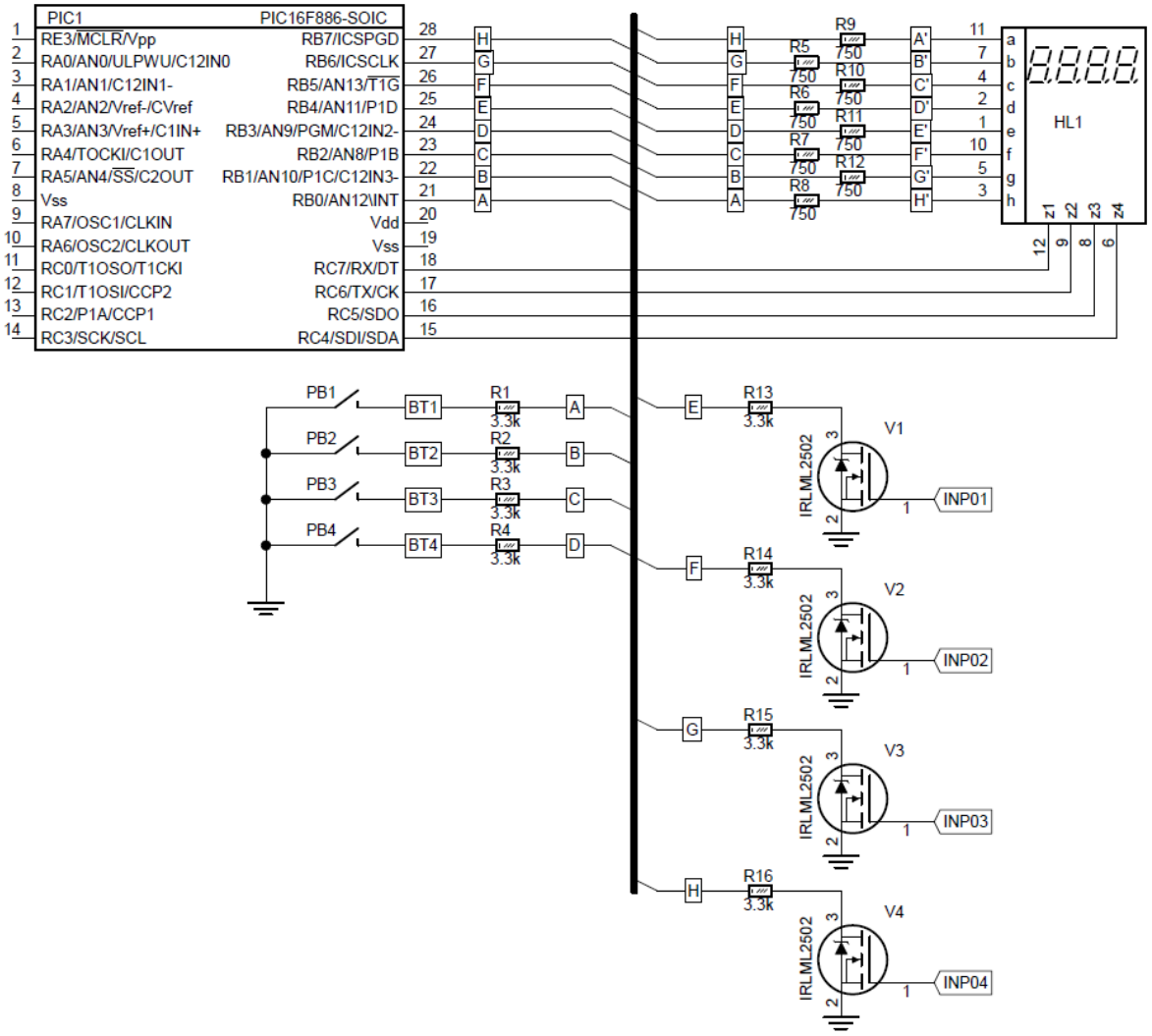
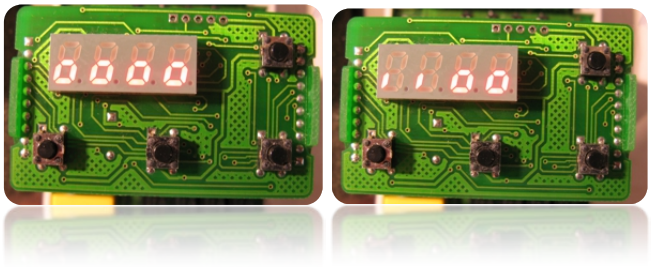


Figure 3

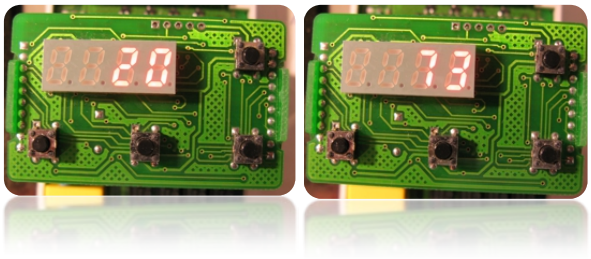
In the demonstration program to display the button is pressed using the principle of counting pulses. It implements the algorithm and then pressing a single auto-dialing with increasing speed by holding down.

Scheme for testing is shown in Figure 3. Resistor value relatively to supply 5 volts. To supply the 3.3 volts necessary resistors in the circuit setodiodnogo indicator to reduce to 220 ohm resistors in the circuit and buttons, and field-effect transistors to increase 6,8-10 k.

To demonstrate the signal from a remote sensors used indication of "little zeros" in the lower segments of the indicator.



To demonstrate the calculation of the pulse - the digital value of the pulse counter.



Watch the video ...

The demo program ...

The version v2.0.
Email: gena.chernov@gmail.com
http://svetomuzyka.narod.ru/ http://invent-systems.narod.ru/