```
// RPM Measurement Using By-Pic BV504   ( Save No 6  )
// ==================================
//  RPM measurement on GPIO port RA8
//  Option to count over 10 revolutions (key #5)
//             or over 100 revolutions (key #6)
//  Key #7 is Cancel  and  Key #14 is Enter
//  RPM calculated and displayed on LCD screen
//  Requires ByPic BASIC loaded together with
//     the ByPic Roukie3 Part 1 & 2 and I/O Routines.

// Global Variables
// ================
//  Global Action variables
dim action


// Program Functions
//*****************

function wait_enter()
   while keystat(0) = 0 ; wend  // wait for key
   while keystat(0) <> 0 ; wend
   wait(100)
endf

function initialise()
// Keypad initialisation
    cap_init()
    cap_start()
    // set up timer1 to go off every 10ms, this will then call
    // ir_touch which fills the keystat() array with values
    tmr_init(*TIMER1(),1,50000) // 10 ms
    ir_set(*TIMER1(),3,"ir_touch")

// Keypad initialisation
    lcd_init()
    wait(500)

// Initialse TIMER23 for use by revs_per_minute
    tmr_init(*TIMER23(),7,0)
    io_pinRole(*RA8(),IN,WPU) // set RA8 as input
    action = 0
endf

function wait_low_high(n)
  // Waits for n succesive Highs on RA8
  //    so as to ignore any jitter on low to high
  dim j,v
  do
    v = 0
    for j = 1 to n ; v = v + io_pinGet(*RA8()) ; next
  until v = n
 endf

function wait_high_low(n)
  // Waits for n succesive Lows on RA8
  //    so as to ignore any jitter on high to low
  dim j,v
  do
    v = n
    for j = 1 to n ; v = v - io_pinGet(*RA8()) ; next
  until v = n
 endf
```

```
function revs_per_minute(n)  // Counts over n revolutions
  // Pulses on input RA8
  dim count,k=50,t1,t2,ms,rpm
  lcd_cls
  lcd_txt("Measuring RPM")
  lcd_rc(1,0)
  lcd_txt(" over "+n+" rev's")
  count = 0
     // Ensure timeing allways starts from same point
     wait_low_high(k)    // Wait for Low to High
     wait_high_low(k)    // Wait for High to Low
  t1=tmr_get(*TIMER23())/156.25 // Get start time ms
  while count< n
     wait_low_high(k)
     wait_high_low(k)
     count = count + 1
     lcd_rc(0,10) ; lcd_txt("   "+count)
  wend
  t2=tmr_get(*TIMER23())/156.25 // Get finish time ms
  ms = t2-t1                    // Difference in millisecs
  rpm = count * 60000 / ms      // Calculate RPM
  lcd_cls
  lcd_txt(ms+" ms for "+count)
  lcd_rc(1,0)
  lcd_txt(" = "+rpm +" rpm")
  lcd_rc(1,13)
  lcd_txt("Ent")
  wait_enter()
endf


//  Get Required Action
//  ******************

function get_action()
   // Reads keypad to determine the required action

  dim key=0, act=-1, prev_act=-1, act$=" ", r$=" ", exit = 0

      lcd_cls ; lcd_txt("RPM Measurement")
      lcd_rc(1,5); lcd_txt("Ready")

    while exit = 0
       // wait for key to be pressed
        while keystat(0) = 0 ; wend
        key = keystat(1)

        select(key)
          case(5) ; act = 5 ; act$ = "Revs/Minute 10"
                      // Count over 10 revolutions

          case(6) ; act = 6 ; act$ = "Revs/Minute 100"
                    // Count over 100 revolutions

          case(7) ; if act > 0 then
                      lcd_rc(1,0);lcd_txt("    Cancelled    ")
                    endif
                    act = 0 ; exit = 1
                    break

          case(14) ; if act > 0 then
                      lcd_rc(1,0);lcd_txt("    Confirmed    ")
```

```
                     endif
                     exit = 1
                     break
          default
            if act <> prev_act then
              prev_act = act
              lcd_cls;lcd_txt(act$)
              lcd_rc(1,0);lcd_txt("ENTER to Confirm")
            endif
         endselect
     wend
     while keystat(0) <> 0 ; wend
  return act
endf


 // Main Function
 //*************

function main()
    // Main Function - runs automatically at start up.

    initialise() ; lcd_cls ; lcd_txt("RPM Measurement")
                lcd_rc(1,0); lcd_txt("    Version 1.0")
    wait(1000)

  while comkey?(2) = 0 // Allows Terminal to stop program
   // Check for Action Keys on Keypad
      action = get_action()
      select(action)
        case(5) ; revs_per_minute(10) // 10 revolutions
        case(6) ; revs_per_minute(100) // 100 revolutions
      endselect
  wend
  lcd_cls ; lcd_txt("EXIT from")
  lcd_rc(1,0) ; lcd_txt("RPM Measurement")
endf
```