AN0001: The data transmission and reception by the AUREL XTR-CYP-2.4 module.

The aim of this document is to illustrate the basic procedures that allow the correct use of the 2.4GHz XTR-CYP-2.4 AUR°EL. transceiver module. The compiler Basic BASCOM(1) is used as the programming tool (a demonstration copy of which can be found on the web site at the following address http://www.mcselec.com/), the microcontroller in use is a ATMEL ATMega8. The application diagram is shown at the end of the document.
Such choise make much easier to transport the supplied software the translation being immediate for several microcontrollers.
It is assumed that the following documents, issued by Cypress and available at the following address: http://www.cypress.com/portal/server.pt?space=CommunityPage&control=SetCommunity&CommunityID=209&PageID=259&fid=65&rpn=CYWUSB6935 , have been axamined.

- *WirelessUSB LR(TM) 2.4-GHz DSSS Radio SoC* [shortened hereinafter in DS]
- *Managing Power in WirelessUSB (TM) Systems* [MP]
- *WirelessUSB(TM) LS Firmware Tips and Tricks* [TT]

The module's programming procedures and the explanatory examples, illustrate the usage technique in connection to low consumption leaving, we mean, the device as long as possible switched off. (Power Down).

(1)BASCOM is a trademark registered by MCS Electronics.

XTR-CYP-2.4 AUREL module connection to a microcontroller (see diagram enclosed):

Set up of the microcontroller ports

```
Config Pinb.2 = Output
Nss_port Alias Portb.2
Set Nss_port                      'Slave Select (output)

Config Pinb.3 = Output
Mosi Alias Portb.3
Mosi = 0                          'MOSI (output)
Config Pinb.4 = Input
Miso Alias Portb.4
Miso = 1                          'MISO (input): port in weak pull up

Config Pinb.5 = Output
Clk Alias Portb.5
Clk = 0                           'SCK (output)

Config Pinc.0 = Input
Pactl Alias Pinc.0                'PACTL: external PA state control
Pactl = 1                         'Port in weak pull up

Config Pind.0 = Input
U_rx Alias Portd.0                'UART_RX

Config Pind.1 = Output
U_tx Alias Portd.1                'UART_TX

Config Pind.2 = Output
Npd_port Alias Portd.2
Npd_port = 0                      'Power Down XTR-CYP-2.4 AUREL
```

```
Config Pind.3 = Input
Irq_port Alias Portd.3
Irq_pin Alias Pind.3
Irq_port = 0                     'IRQ: XTR-CYP-2.4 AUREL interrupt

Config Pinc.1 = Output
Nreset Alias Portc.1
Nreset = 1                       'XTR-CYP-2.4 AUREL reset

Config Pinc.5 = Output
Led Alias Portc.5
Led = 0                         'Led (output)
```

The Nreset value is initialized to 1: in this way the XTR-CYP-2.4 AUREL is immediately active. Through this line there is the possibility to bring back the registers to the initial default setting decided by the Manufacturer of the RF chip.

The SPI routines

In [DS] is reported how the programming modality of the RF module is through SPI connection with a speed not more than 2Mbit.
The microcontroller involved in these examples must be therefore programmed in a way to take advantage of the 8MHz internal oscillator, programming the hardware SPI port at a speed of 2Mbit, the maximum allowed, to minimize the times.
The programming language BASCOM in which these examples are written, operates the SPI communication by restricted commands. By combining these SPI commands with what is demanded by the RF unit [DS] Manufacturer, are obtained these two routines which form the dialogue base with the XTR-CYP-2.4 AUREL. At the beginning of the program the SPI connection must be enabled by the following commands:

```
Config Spi = Hard , Master = Yes , Polarity = Low , Phase = 0 , Clockrate = 4 , Noss = 1
Spiinit
```

The following routine enables to write a value ***Vaalue*** in the register ***Aaddr.*** The BASCOM needs the two data in a buffer (or in memory's adjacent locations). The Nss port must commutate before and after the data transmission procedure [DS].

```
SPI WRITING
Sub Wspi(byval Aaddr As Byte , Byval Vaalue As Byte)
   Dim W(2) As Byte

   W(1) = Aaddr
   W(2) = Vaalue

   W(1) = W(1) Or &B10000000
   Reset Nss_port
   Spiout W(1) , 2
   Set Nss_port
End Sub
```

The reading function foresees a first transmission of the XTR-CYP-2.4 AUREL register's address to be read, then the 0x00 transmission: the outgoing data on the MOSI line is meaningless, it serves only, according to SPI protocol, to have a clock generated by the Master unit for the outgoing data from the Slave on the MISO line. The data read by the microcontroller, is automatically saved in the SPDR register that will be read by the following program and available in the output as function's value.

```
SPI  READING
Function Rspi(byval Aaddr As Byte) As Byte
   Dim R(2) As Byte

   R(1) = Aaddr
   R(2) = &B00000000

   Reset Nss_port
   Spiout R(1) , 2
   Rspi = Spdr
   Set Nss_port
End Function
```

Exit from the Power Down state (coming from no supply or hardware reset)

The following instructions wake up the XTR-CYP-2.4 AUREL from the Power Down state and load all the parameters required for the operation that will remain permanent available at the following address:

```
Set Npd_port                          'XTR-CYP-2.4 AUR°EL wakeup
Waitus 5000                           'Waiting wakeup time
```

The first instruction wakes up the transceiver, while the second inserts a 5msec delay: this has to be compulsorily inserted only at the first switch on of the system since the handshaking check is not yet on-line. Subsequently to this first switch on, the IRQ line shall be utilized, in order to minimize the operating times, since such signal shall indicate the exit of the XTR-CYP-2.4 AUR°EL from the Power Down state.
We shall now check the presence of the device in the connector (see explanatory diagram in the end of the document) and its correct opertion:

```
Dim Canale As Byte                    'Definition of used variables
Dim Ch As Byte


Ch = Rspi (&H00)                      'Reading register containing silicon revision ID

             If Ch = &HFF Then        'If device isn't on the connector or if it
                                      doesn't operate correctly, due to pull up on the
                                      MISO line, FF is read.
      Do
             Led=1                    'Led flashes with a 1 second frequency. Make sure
                                      that device is correctly present on connector.
             Waitms 500
             Led=0
             Waitms 500
      Loop
Else
      Portb.4=0                       'If device is present and it operates correctly
                                      pull up on MISO line is removed to reduce
                                      consumption.
End If

      If Pactl = 1 Then               'Control, similar to previous one, of external PA
                                      state.

             Led=1                    'Led flashes with a 2 seconds frequency
             Wait 1
             Led=0
             Wait 1
      Loop
```

```
Else
    Portc.0 = 0                        'Pull up is removed to reduce consumption
End If
```

If the device is in place and it works properly, things then may proceed .
The following parameters are now programmed:

SERDES technical enabling of data passing
Enablig of a whole byte for SERDES…………………………(ONLY IN TX)
IRQ modality choise
External PA driving's polarity choise…..................……….(ONLY IN TX)
Disabling of the external clock at 13Mhz
A fix delay choise to show the PLL lock ....……… (ONLY IN TX)
Enabling of the WAKEUP interrupt
Configuration of the output power......………………………..(ONLY IN TX)
RF transmission and reception speed
Validity thresholds of the received signal …………………… (ONLY IN RX)
RF channel number

```
Call Wspi(&H06 , &B00001000)       'SERDES enable

Call Wspi(&H10 , &B11111111)       'Validity of all bits of SERDES register

Call Wspi(&H05 , &B00000001)       'Selection of drive method of IRQ pin: Active
                                   high CMOS

Call Wspi(&H20 , &B00000110)       'Selection of polarity and enable of pin for
                                   control of external power amplifier

Call Wspi(&H24 , &B01000000)       '13MHz clock output disable to avoid sensitivity
                                   reduction (see [DS])

Call Wspi(&H38 , &H7D)             '250µs fixed delay between TX enable and preamble
                                   sending

Call Wspi(&H1C , &B00000001)       'Wakeup interrupt enable: a wakeup event is
                                   triggered when IRQ line is asserted.

Ch = Rspi(&H1d)                    'Reading to deassert IRQ line.

Call Wspi(&H23 , &B00000111)       'Choosing of output power in function of first 3
                                   bits of 0x23 register. Available configurations
                                   from 000 to 111 (step from 0 to 7).
```

Step	RF output power (ERP)
0	-14 dBm
1	-10 dBm
2	-6 dBm
3	-1 dBm
4	6 dBm
5	10 dBm
6	13 dBm
7	15 dBm

For the speed selection the three instructions related to 16Kbit/sec or the three releted to 64Kbit/sec must be utilized
If the 16Kbit/sec speed is wanted then utilize the three following lines, if the 64kbit/s speed is wanted than utilize the subsequent three lines.

' For RF transmission speed at 16Kbit/sec :

```
Call Wspi(&H04 , &B00000000)          '16Kbit/sec RF data rate

Call Wspi(&H19 , &B00001000)          'Data discrimination threshold low

Call Wspi(&H1a , &B00111000)          'Data discrimination threshold high
```

' For RF transmission speed at 64Kbit/sec :

```
Call Wspi(&H04 , &B00000110)          '64Kbit/sec RF data rate

Call Wspi(&H19 , &B00000001)          'Data discrimination threshold low

Call Wspi(&H1a , &B00011111)          'Data discrimination threshold high


Channel = 40                          'Channel at 2.440GHz
Call Wspi(&H21 , Channel)
```

The previous parameters programming must be carried out once only at the device switch on (or in case of no power supply or hardware reset).
Now we can proceed whether leaving the device active or by placing it in Power Down state for a subsequent timed wake up in order to minimize the consumptions.

In our case we place the device in Power Down state and operate a consequent wake up.

```
Reset Npd_port                        'Device in Power down
```

The device is now waked up and set under reception.

PROGRAMMING OF THE XTR-CYP-2.4 AUR°EL UNDER RECEPTION

The device is placed in stand-by awaiting for a data packet of undefined length, with a consequent possible menagement , and final switching off of the device.

Under reception the device effectivelly behaves as an UART by signalling the presence of a valid byte which has to be read by a proper command. It is possible to enable a flag that signals the end of the bytes under reception, by evaluating a programmable Time out in position to discriminate a time without data reception. Since such information is on the same line of the valid datum's presence and, not supposing to receive data packets of known length, we prefer not to utilize the EOF (End Of Frame) information but a Time out software that, at the expiring of 1,5 byte without reception, asserts the entered packet.

```
Dim Datain (255) As Byte                      '255 bytes receiver buffer

Dim Numdatain As Byte                         'Received bytes counter

Dim Ch As Byte                                'Received byte
```

```
Dim F_ovf As bit                          'End of packet flag

Dim T_bit As Byte

Numdatain = 0                             'Reset of data counter

F_ovf = 0                                 'Reset of end of packet flag
Config Timer0 = Timer , Prescale = 64     'Counter that increase every 8µs
Stop Timer0
On Ovf0 Tc0_end_packet                    'In case of Overflow it calls
                                          Tc0_end_packet

T_bit =165                                'Overflow after 750µs for 16Kbit/sec RF
                                          data rate. If 64Kbit/sec is used it is
                                          necessary to fix T_bit=230 to have overflow
                                          after 200µs.
Enable Interrupts
Enable Timer0


Set Npd_port                              'Wakeup

While Irq_pin = 0                         'Waiting loop for device wakeup: when a
                                          wakeup event is triggered IRQ line is
                                          asserted and there isn't delay to exit from
                                          loop

Ch = Rspi(&H1d)                           'By reading wakeup status register IRQ line
                                          is deasserted.

Call Wspi(&H03 , &B10000000)              'RX ENABLE

Waitus 250                                'Waiting for RX stable

Ch = Rspi(&H09)                           'First byte is read in case it is yet
                                          present but not necessary correct.

Call Wspi(&H07 , &B00000001)              'FULL interrupt Enable: IRQ line is
                                          asserted when a complete byte is received.

Do
  If Irq_pin = 1 Then                     'Every received byte causes IRQ line low-
                                          high transition.

      Stop Timer0                         'Received byte. Time Out reset
      Timer0 = T_bit

      Numdatain = Numdatain + 1           'Numdatain counter must be initialized to 0
                                          so to be incremented to 1 when first byte
                                          is received.

     Datain(Numdatain) = Rspi(&H09)       'Reading of byte present into receive data
                                          register

    Start Timer0                          'Timer switches at the reception of every
                                          byte

Else
       If F_ovf = 1 Then
              Goto Valid_packet
       End if
```

```
End If

Loop


Valid_packet:

'the entered packet length is Numdatain byte
'Possible management of the packet

Reset Npd_port                              'Device in Power down


'Timeout at the end of the packet

Tc0_end_packet:

      Stop Timer0
      F_ovf = 1

Return
```

PROGRAMMING OF THE XTR-CYP-2.4 AUR°EL UNDER TRANSMISSION

The device is activated, it transmits a single packet, NUMDATAOUT = 20 byte long, loaded in the DATAOUT transmission baffer, and set back to Power Down.

```
Dim Dataout(20) As Byte                     '20 byte transmission buffer

Dim Numdataout As Byte                      'Counter of number of byte trasmitted
Dim J As Byte

Numdataout = 20

   Dataout(1) = &H30                        'Loading of transmission buffer
   Dataout(2) = &H31
   Dataout(3) = &H32
   Dataout(4) = &H33
   Dataout(5) = &H34
   Dataout(6) = &H35
   Dataout(7) = &H36
   Dataout(8) = &H37
   Dataout(9) = &H38
   Dataout(10) = &H39
   Dataout(11) = &H41
   Dataout(12) = &H42
   Dataout(13) = &H43
   Dataout(14) = &H44
   Dataout(15) = &H45
   Dataout(16) = &H46
   Dataout(17) = &H47
   Dataout(18) = &H48
   Dataout(19) = &H49
   Dataout(20) = &H4A

'The packet to be sent is "0123456789ABCDEFGHIL"


Set Npd_port                                'Wakeup
```

```
While Irq_pin = 0                  'Waiting loop for device wakeup: when a
                                   wakeup event is triggered IRQ line is
                                   asserted and there isn't delay to exit from
                                   loop
Wend

Ch=Rspi(&H1d)                      'By reading wakeup status register IRQ line
                                   is deasserted.

Call Wspi(&H0d , &B00000001)       'Empty interrupt enable: when transmission
                                   register is empty IRQ line is asserted and
                                   it is so possible reload the register

Call Wspi(&H0F , Dataout(1))       'First byte to send is loaded into
                                   transmission register
```

'By this trick [TT] we load the data transmission register before to enable the transmission itself: when, subsequently, the transmission will be enabled, the XTR-CYP-2.4 AUR°EL shall transmit a synchronism preamble and the datum ready available in the register.

```
Call Wspi(&H03 , &B01010000)       'TX ENABLE.XTR-CYP-2.4 AUR°EL is programmed
                                   in Fixed Syn Lock Mode: device starts to
                                   transmit preamble and data packet with a
                                   fixed delay, programmed into register 0x38
                                   and in our case equal to 250µs, after TX
                                   Enable command.

Do
Loop Until Irq_pin = 1             'Waiting loop: when transmission register
                                   is empty IRQ line is asserted and it is so
                                   possible loading a new byte

Numdataout = Numdataout-1

For J = 2 To Numdataout

      Call Wspi(&H0F , Dataout(j)) 'Byte sending
      Do                           'Waiting loop: when transmission register
                                   is empty IRQ is asserted and it is so
                                   possible to load the new byte
      Loop Until Irq_pin = 1
Next J

Numdataout = Numdataout+1

Call Wspi(&H0F , Dataout(Numdataout))  'Last byte sending: it isn't necessary to
                                   wait Empty Interrupt because no new byte
                                   need to be loaded.

Call Wspi(&H0d , &B00000010)       'Empty interrupt disable and Done interrupt
                                   Enable: IRQ line is asserted when last byte
                                   sending is terminated. It is so possible
                                   to put XTR-CYP-2.4 AUR°EL in Power Down.

Do
Loop Until Irq_pin = 1             'Waiting  for IRQ line asserting

Reset Npd_port                     'Device in Power Down
```

APPLICATION EXAMPLES OF THE XTR-CYP-2.4 AUR°EL MODULE

Here below an example of the RF data reception software and forward to serial when the end of the RF packet is identified by means of the time out technique previously described.
The device is always activated

RX.BAS

```
'-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
'Example of the RF reception software and serial forwarding
'-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*

$regfile = "m8def.dat"
$crystal = 8000000
$baud = 19200


'-*-*-*-*-*-*-*-*-*-*-*-*-
'Variables Definition
'-*-*-*-*-*-*-*-*-*-*-*-*-

Dim Datain(255) As Byte
Dim Numdatain As Byte
Dim Cnttx As Byte
Dim Endbyte As Bit
Dim T_bit As Byte
Dim J As Byte
Dim Canale As Byte
Dim Ch As Byte


'-*-*-*-*-*-*-*-*-*-*-*-*-
'Interface procedure
'-*-*-*-*-*-*-*-*-*-*-*-*-

Declare Sub Send_data()
Declare Sub Wspi(byval Aaddr As Byte , Byval Vaalue As Byte)
Declare Function Rspi(byval Aaddr As Byte) As Byte
Declare Sub Reception()
Declare Sub Reset_device()


'-*-*-*-*-*-*-*-*-*-*-*-*-*-
'Interrupt configuration
'-*-*-*-*-*-*-*-*-*-*-*-*-*-

Config Timer0 = Timer , Prescale = 64
On Ovf0 Tc0_end_packet
Stop Timer0

On Utxc Uart_tx
On Udre Uart_dre
Disable Utxc
Disable Udre


'*-*-*-*-*-*-*-*-*-*-*-*-*-*-**-*-*-*
'*-*   Alias PORTS and PIN  *-*
'*-*-*-*-*-*-*-*-*-*-*-*-*-*-**-*-*-*
```

```
Irq_pin Alias Pind.3
Irq_port Alias Portd.3
Led Alias Portc.5
Nss_port Alias Portb.2
Npd_port Alias Portd.2
Pactl Alias Pinc.0

'******** RESET *********

Call Reset_device


Config Spi = Hard , Master = Yes , Polarity = Low , Phase = 0 , Clockrate = 4 , Noss = 1

Spiinit

Led = 0

Set Npd_port                  'Switch on Cypress

Waitus 5000

Ch = Rspi(&H00)

If Ch = &HFF Then
   Do
      Led = 1
      Waitms 500
      Led = 0
      Waitms 500
   Loop
Else
   Portb.4 = 0
End If

If Pactl = 1 Then
   Do
      Led = 1
      Wait 1
      Led = 0
      Wait 1
   Loop
Else
   Portc.0 = 0
End If


Call Wspi(&H06 , &B00001000)  'SERDES enabled     Reg_serdes_ctl   0x06
Call Wspi(&H05 , &B00000001)  'IRQ active         Reg_config       0x05
Call Wspi(&H24 , &B01000000)  'Disabled 13MHz     Reg_xtal_adj     0x24
Call Wspi(&H1c , &B00000001)  'Enable wake up interrupt

Ch = Rspi(&H1d)


'RF speed selection, related thresholds and timer

'16Kbit/sec
Call Wspi(&H04 , &B00000000)  '64 bit                       Reg_data_rate    0x04
Call Wspi(&H19 , &B00001000)  'Low threshold 0x08           Reg_Threshold_L  0x19
Call Wspi(&H1a , &B00111000)  'High threshold 0x38          Reg_Threshold_H  0x1A
T_bit = 165


'64Kbit/sec
'Call Wspi(&H04 , &B00000110)  '32 bit Double Data Rate     Reg_data_rate    0x04
'Call Wspi(&H19 , &B00000001)  'Low threshold               Reg_Threshold_L  0x19
'Call Wspi(&H1a , &B00011111)  'High threshold              Reg_Threshold_H  0x1A
```

```
'T_bit = 230


Channel = 40

Call Wspi(&H21 , Channel)


Numdatain = 0

Enable Interrupts

Enable Timer0
Stop Timer0


Call Wspi(&H03 , &B10000000)  'RX ENABLE           Reg_control      0x03

Waitus 250

Ch = Rspi(&H09)

Call Wspi(&H07 , &B00000001)  'Enabling of the Full interrupt



'*-*-*-*-*-*-*-*-*-*-*-*
'*-*   Reception   *-*
'*-*-*-*-*-*-*-*-*-*-*-*


Do
   If Irq_pin = 1 Then
      Stop Timer0
      Timer0 = T_bit
      Numdatain = Numdatain + 1
      Datain(numdatain) = Rspi(&H09)
      Start Timer0
   End If
Loop



'*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
'*-*    Timer at the end of the packet     *-*
'*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*

Tc0_end_packet:
Stop Timer0


   Enable Interrupts

   Call Send_data

   J = Rspi(&H08)

   Numdatain = 0


Return


'*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
'*-*   Serial transmission   *-*
'*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
```

```
Sub Send_data

   Endbyte = 0
   Cnttx = 1

   Udr = Datain(cnttx)
   Enable Udre


   While Endbyte = 0
   Wend

End Sub


'-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
'  Interrupt on serial transmission
'-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*

Uart_dre:

   If Cnttx = Numdatain Then
      Disable Udre
      Enable Utxc
   Else
      Cnttx = Cnttx + 1
      Udr = Datain(cnttx)
   End If

Return



Uart_tx:

   Disable Utxc

   Endbyte = 1

Return


'*-*-*-*-*-*-*-*-*-*-*-*-*
'   SPI  writing
'*-*-*-*-*-*-*-*-*-*-*-*-*

Sub Wspi(byval Aaddr As Byte , Byval Vaalue As Byte)
   Dim W(2) As Byte

   W(1) = Aaddr
   W(2) = Vaalue

   W(1) = W(1) Or &B10000000
   Reset Nss_port
   Spiout W(1) , 2            'write 2 bytes aw(1), aw(2)
   Set Nss_port
End Sub


'*-*-*-*-*-*-*-*-*-*
'   SPI reading
'*-*-*-*-*-*-*-*-*-*

Function Rspi(byval Aaddr As Byte) As Byte
   Dim R(2) As Byte

   R(1) = Aaddr
```

```
   R(2) = &B00000000

   Reset Nss_port
   Spiout R(1) , 2
   Rspi = Spdr
   Set Nss_port
End Function

'*-*-*-*-*-*-*-*-*-*-*
'   Micro reset
'*-*-*-*-*-*-*-*-*-*-*

Sub Reset_device()


   Config Pinb.2 = Output
   Set Nss_port                'Slave Select output

   Config Pinb.3 = Output
   Mosi Alias Portb.3
   Mosi = 0                    'MOSI output

   Config Pinb.4 = Input
   Miso Alias Portb.4
   Miso = 1                    'MISO input

   Config Pinb.5 = Output
   Clk Alias Portb.5
   Clk = 0                     'SCK output

   Config Pinc.0 = Input
   Portc.0 = 1                 'External PA state: check pin

   Config Pinc.1 = Output
   Nreset Alias Portc.1
   Nreset = 1                  'Cypress' reset

   Config Pinc.5 = Output
   Led = 0                     'Led output

   Config Pind.0 = Input
   U_rx Alias Portd.0          'Cypress RX RF


   Config Pind.1 = Output
   U_tx Alias Portd.1          'Cypress TX RF

   Config Pind.2 = Output
   Npd_port = 0                'Cypress' Power Down


   Config Pind.3 = Input
   Irq_port = 0                'Cypress' interrupt pin

End Sub

End                            'end program
```

Here below an example of the software of a cyclic transmission of a data packet composed by the string "0123456789ABCD-" followed by an incremental number from 0 to 999 and by <CR> and <LF>.
The device is switched on, inizialized and set in Power Down.
Subsequently, in a cycle way (with 300ms clock rate) it is waked up, the packet transmitted and againt set back to Power Down.
The transmission procedure is carried out following what stated above with the aim to keep the device switched on as less time as possible in order to minimize the power consumption.

TX.BAS

```
'-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
'Example of the software for RF cyclic transmission
'-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*

$regfile = "m8def.dat"
$crystal = 8000000
$baud = 19200


'-*-*-*-*-*-*-*-*-*-*-*-*-
'Variables definition
'-*-*-*-*-*-*-*-*-*-*-*-*-

Dim Dataout(255) As Byte
Dim Numdataout As Byte
Dim J As Byte
Dim Canale As Byte
Dim Ch As Byte
Dim Pck_unit As Byte
Dim Pck_dec As Byte
Dim Pck_cent As Byte


'-*-*-*-*-*-*-*-*-*-*-*-*-
Interfacia procedure
'-*-*-*-*-*-*-*-*-*-*-*-*-

Declare Sub Send_rf()
Declare Sub Wspi(byval Aaddr As Byte , Byval Vaalue As Byte)
Declare Function Rspi(byval Aaddr As Byte) As Byte
Declare Sub Reset_device()


'*-*-*-*-*-*-*-*-*-*-*-*-*-**-*-*-*
'*-*   Alias PORTS and PIN   *-*
'*-*-*-*-*-*-*-*-*-*-*-*-*-**-*-*-*

Irq_pin Alias Pind.3
Irq_port Alias Portd.3
Led Alias Portc.5
Nss_port Alias Portb.2
Npd_port Alias Portd.2
Pactl Alias Pinc.0


'******** RESET *********

Call Reset_device


Config Spi = Hard , Master = Yes , Polarity = Low , Phase = 0 , Clockrate = 4 , Noss = 1

Spiinit

Led = 0

Pck_unit = &H30
Pck_dec = &H30
Pck_cent = &H30

'*-*-*-*-*-*-*-*-*-*-*-*-**-*-*-*
'*-*   Initialization   *-*
'*-*-*-*-*-*-*-*-*-*-*-*-**-*-*-*

Set Npd_port                'Switch on Cypress
Waitus 5000
```

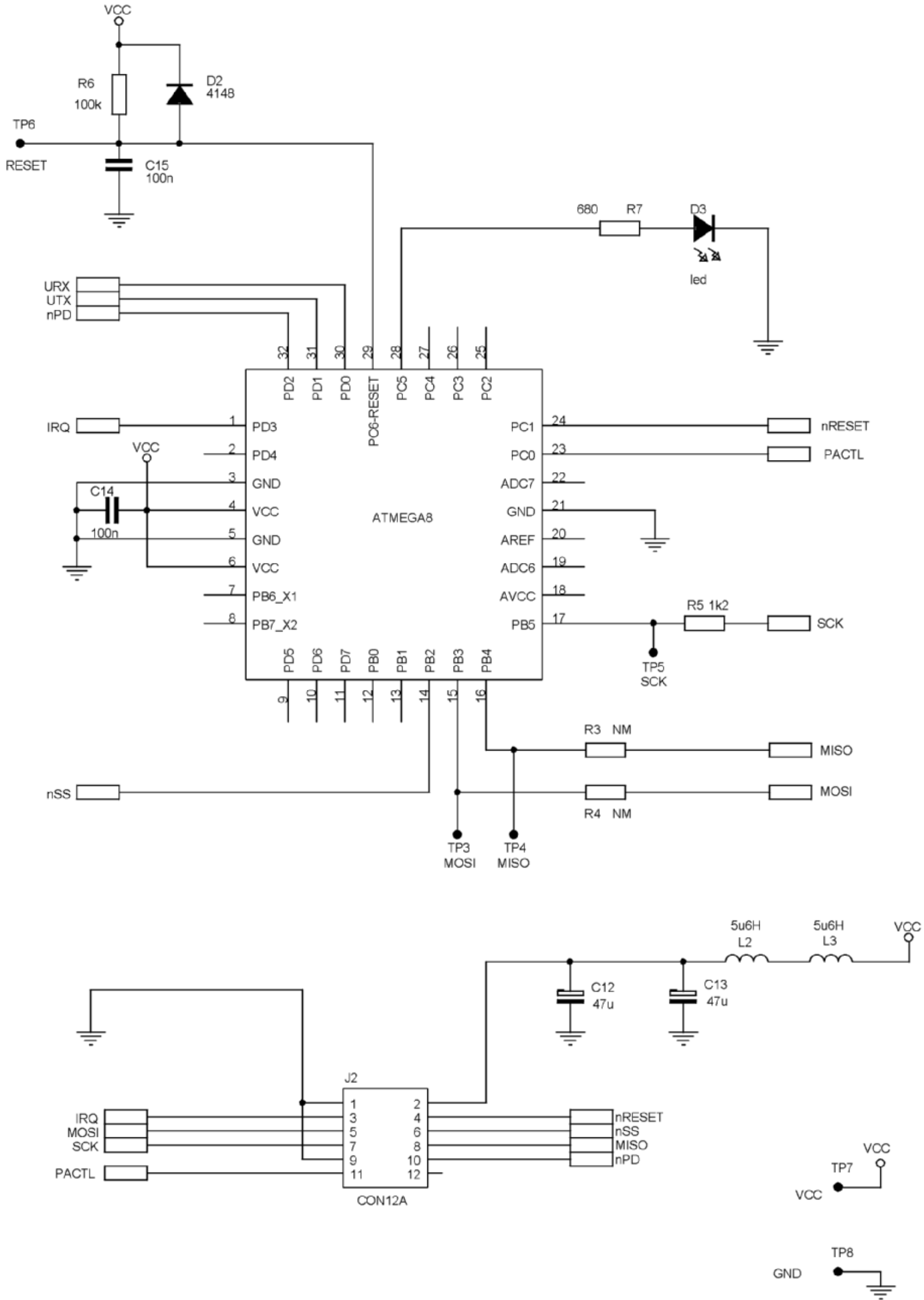
```
Ch = Rspi(&H00)

If Ch = &HFF Then

   Do
      Led = 1
      Waitms 500
      Led = 0
      Waitms 500
   Loop
Else
   Portb.4 = 0
End If


If Pactl = 1 Then
   Do
      Led = 1
      Wait 1
      Led = 0
      Wait 1
   Loop
Else
   Portc.0 = 0
End If


Call Wspi(&H06 , &B00001000)  'SERDES enabled     Reg_serdes_ctl   0x06
Call Wspi(&H10 , &B11111111)  'Tx Valid           Reg_tx_valid     0x10
Call Wspi(&H05 , &B00000001)  'IRQ drive          Reg_config       0x05
Call Wspi(&H20 , &B00000110)  'PA OUTPUT
Call Wspi(&H24 , &B01000000)  'Disabled  13MHz    Reg_xtal_adj     0x24
Call Wspi(&H38 , &H7D)
Call Wspi(&H1c , &H01)        'Enable wakeup interrupt
Ch = Rspi(&H1d)
Call Wspi(&H23 , &B00000111)  'Power last 3 bit   Reg_pa           0x23


'Choise according to intended speed

'16Kbit/sec
Call Wspi(&H04 , &B00000000)  '64 bit                      Reg_data_rate    0x04


'64Kbit/sec
'Call Wspi(&H04 , &B00000110)  '32 bit Double Data Rate     Reg_data_rate    0x04


Channel = 40
Call Wspi(&H21 , Channel)

Reset Npd_port


'*-*-*-*-*-*-*-*-*-*-*-*-**-*
'*-*   Transmission    *-*
'*-*-*-*-*-*-*-*-*-*-*-*-**-*

Do

Numdataout = 20

Dataout(1) = &H30
Dataout(2) = &H31
Dataout(3) = &H32
Dataout(4) = &H33
Dataout(5) = &H34
```

```
Dataout(6) = &H35
Dataout(7) = &H36
Dataout(8) = &H37
Dataout(9) = &H38
Dataout(10) = &H41
Dataout(11) = &H42
Dataout(12) = &H43
Dataout(13) = &H44
Dataout(14) = &H45
Dataout(15) = &H2D
Dataout(16) = Pck_cent
Dataout(17) = Pck_dec
Dataout(18) = Pck_unit
Dataout(19) = &H0D
Dataout(20) = &H0A

Pck_unit = Pck_unit + 1

If Pck_unit = &H3A Then
   Pck_unit = &H30
   Pck_dec = Pck_dec + 1
   If Pck_dec = &H3A Then
      Pck_dec = &H30
      Pck_cent = Pck_cent + 1
      If Pck_cent = &H3A Then
         Pck_cent = &H30
      End If
   End If
End If


   Set Npd_port

   While Irq_pin = 0
   Wend

   Ch = Rspi(&H1d)

   Call Send_rf

   Call Wspi(&H03 , &H00)      'TX and RX disabled

   Reset Npd_port

   Waitms 300

Loop



'*-*-*-*-*-*-*-*-*-*-*-**-*-*
'*-*   RF Transmission   *-*
'*-*-*-*-*-*-*-*-*-*-*-**-*-*

Sub Send_rf


   Call Wspi(&H0d , &B00000001)       'Empty interrupt enabled

   Call Wspi(&H0f , Dataout(1))       'Reg_tx_data  0x0f

   Call Wspi(&H03 , &B01010000)       'Reg_control    0x03  TX ENABLE

   Do
   Loop Until Irq_pin = 1


   Numdataout = Numdataout - 1
```

```
   For J = 2 To Numdataout

         Call Wspi(&H0f , Dataout(j))       'Reg_tx_data  0x0f
         Do
         Loop Until Irq_pin = 1

    Next J

    Numdataout = Numdataout + 1

    Call Wspi(&H0f , Dataout(numdataout))

    Call Wspi(&H0d , &B00000010)       'Done interrupt enabled

    Do
    Loop Until Irq_pin = 1


End Sub



'*-*-*-*-*-*-*-*-*-*-*-*
'   SPI writing
'*-*-*-*-*-*-*-*-*-*-*-*

Sub Wspi(byval Aaddr As Byte , Byval Vaalue As Byte)
   Dim W(2) As Byte

   W(1) = Aaddr
   W(2) = Vaalue

   W(1) = W(1) Or &B10000000
   Reset Nss_port
   Spiout W(1) , 2              'write 2 bytes aw(1), aw(2)
   Set Nss_port
End Sub


'*-*-*-*-*-*-*-*-*-*
'   SPI reading
'*-*-*-*-*-*-*-*-*-*
Function Rspi(byval Aaddr As Byte) As Byte
   Dim R(2) As Byte

   R(1) = Aaddr
   R(2) = &B00000000

   Reset Nss_port
   Spiout R(1) , 2
   Rspi = Spdr
   Set Nss_port
End Function



'*-*-*-*-*-*-*-*-*-*-*-*
'   Micro reset
'*-*-*-*-*-*-*-*-*-*-*-*

Sub Reset_device()


   Config Pinb.2 = Output
   Set Nss_port                 'Slave Select oitput

   Config Pinb.3 = Output
   Mosi Alias Portb.3
```

```
   Mosi = 0                     'MOSI output

   Config Pinb.4 = Input
   Miso Alias Portb.4
   Miso = 1                     'MISO input

   Config Pinb.5 = Output
   Clk Alias Portb.5
   Clk = 0                      'SCK output

   Config Pinc.0 = Input
   Portc.0 = 1                  ''External PA state: check pin

   Config Pinc.1 = Output
   Nreset Alias Portc.1
   Nreset = 1                   'Cypress reset

   Config Pinc.5 = Output
   Led = 0                      'Led output

   Config Pind.0 = Input        'Cypress RX RF
   U_rx Alias Portd.0

   Config Pind.1 = Output       'Cypress TX RF
   U_tx Alias Portd.1

   Config Pind.2 = Output
   Npd_port = 0                 'Cypress Power Down

   Config Pind.3 = Input        'Cypress interrupt pin
   Irq_port = 0


End Sub

End                             'end program
```

Connections diagram between microcontroller and XTR-CYP-2.4 AUR°EL



AN0002

AN0002: APPLICATION EXAMPLES FOR XTR-CYP-2.4 AUREL MODULE AIMED AT LOW CONSUMPTION

The application example given with the software indicated in the Application Note AN0001, foresees that the receiver is always switched on, the transmitter, instead, is activated only for the time strictly required for the data transmission.
This application is a generic one and, therefore, in order to emphasize the real possibilities of the system, we report now, here below, further possible solutions in particularly important cases, aimed at the low consumption of the device. The RF side consumption only shall be emphasized giving to the User the calculation of the additional consumptions related to micro, display, led etc.

Remote control for manufacturing machines, mobile arm systems, cranes

This case includes the mobile remote controls (activated by an Operator) which require an immediate transmission of the information and a data response able to enhance peculiar (particular) conditions taken up by the controlled systems in motion.
The receiving system is considered continuously powered since it is integral with the crane's body, mobile system frame etc.
The quantity of the data to be transmitted and received is of 8 bytes suitable to contain a 32bit code plus 4 bytes of data (typically the operating push-buttons).
In order to let the Operator obtain the immediateness of the order accomplishment, the data are transmitted every 30ms during the whole time in which the push-buttons are pressed.
Supposing an answer of 8 bytes per frame, with a speed of 64Kb/S, we will have:

3mS time of TX/RX per frame.	The theoretic time is 1+1 ms increased to 3 for latency times
Average consumption of 80mA over the 3mS.	Average between 100mA TX and 60mA RX
Continue utilization for 4 hrs/day	Over 8 hrs/day of work the estimated operation time is the 50%.

The 3ms time over 30mS reduces the consumption to 8mA.
4hrs over 24hrs reduce by factor 6 the consumption = 1.33 mA

This type of remote control normally employs a rechargeable batteries or at least of 1000mA /h capacity.
In the formulated hypothesis the working time shall be of 1000/1.33 = 750 hrs = 31days.
The sole radiofrequency allows then a transmission and reception of commands at high safety for a month time; an extremely favourable condition.

Time thermostat, temperature sensors transmission with return of confirmation data

From a remote position a sensor's information (typically a temperature) has to be transmitted.
Return data have to be foreseen for confirmation as well as for command (e.g. information display, operation times variation etc).
The receiving system is considered always powered (boiler, sensors supervisor) and in the sensors case the reception up to 32 different sensor is estimated.
Alarm conditions or manual interventions excepted, a transmission and reception each 30 seconds are estimated.
3mS of TX/RX per frame
As above the average consumption is 80mA but each 30Sec.
The average consumption becomes, therefore, equal to 8uA which can be increased up to 10uA considering a 20% account for alarms or manual operations.

Considering a duty cycle so low, the installation of several sensors scattered in the same area does not trouble since the collision probability is about 1/300 and, in the temperature sensors case, this loss of data is estimated bearable.
Considering to employ non rechargeable batteries of 700mA/h capacity (2 AAA type utilized up to half power) for the sole radio frequency they will last for 70.000 hrs equal to more than 7 years.

AUREL S.p.A. Via Foro dei Tigli, 4 - 47015 Modigliana (FC) – ITALY
Tel.: +39.0546.941124 Fax: +39.0546.941660
http://www.aurelwireless.com - email: lab-el@aurel.it

The nickel-cadmium batteries do not last for so long time and this foster further hypothesis:

1) It is decided to transmit at a low speed and at the maximum distance by employing the 16Kbsec.
   In this case all the RF utilization times quadruple bringing the batteries duration to less than 2 years. For nickel-cadmium batteries this is considered acceptable.
2) The intervention time can be halved (15 Sec) bringing thus the duration to 3.5 years.
   Interesting hypothesis if the sensors' application technique requires such type of control time.

Mobile systems' safety

A device in motion has to be controlled by a radio frequency link active the whole motion time so to guarantee (in case of RF failure) the maximum safety against noises, battery discharge etc.
The device is equipped with a receiver active at a low duty cycle, waked up from a control workstation. Once the communication is hooked up the system operates with a transmission and a reception at 60mS of cycle so to guarantee a quick block in case of lack of consent.

Hypothesis:

1. Receiver switched on for 500uS over 1000mS
2. 3mS of TX/RX over 60mS for one motion cycle of 50 s
3. 30 movements per day.

The first hypothesis is feasible by utilizing the 64Kbsec that became thus essentials.
Consumption = 60mA /2000 = 30uA
The RX/TX cycle of 3mS over 60mSec entails 80mA/20 = 4000uA . Considering 1500s of daily movements, and knowing that one day contains 86400 seconds, the average consumption becomes 4000uA*(1500/86400) = 70uA
The total consumption, therefore, is 30+70 = 100uA.

A typical movement of this example is a motorized gate that requires a check during its motion. The first second is employed to wake up the RX device and to start the controlled motion cycle (50sec). In case of link loss (safeties activated by obstacles to the motion) the system can stop in a 60ms reaction time, which is absolutely adequate for a slow mechanical movement. The 30 operations per day are reasonable and the 50s time per movement consider the opening-pause-closing.
In these hypothesis the consumption, therefore, is equal to 100uA that, giving a lifetime of at least 2 years, requires a 2A/h battery.
For this reason 3.6V 2A nickel-cadmium or lithium batteries (small torch type) must be employed

Phone-in voting systems

The vote expressed by a number of persons in a room each of them equipped with a palmtop transceiver, must be read.
The master system is always powered and the single transceivers are RX activated by pressing a push-button that indicates the individual vote.
Supposing to monitor max. 1000 persons; let's calculate the consumption and the latency time of the system.
To identify each voter 2 bytes are required; assuming a number over 256, at high-speed transmission, therefore, it would take 250us+250us, (assuming 250us of transmitter settling). To this must follow the remote vote reception that may take place in the following millisecond. Let's take as a safety margin the 3mS value to obtain a question with answer; therefore the whole polling will take 3 sec.
The remote devices must remain alive at least for 3 sec to allow the reading by the master. Let's bring the switch on time to 6 sec. to allow the master a second reading and suppose a vote each 10 minutes for 4 hrs.
6 seconds over 600 seconds give an average consumption of 600uA (only receiver since the TX time is very short) and the use of 4 hrs over 24 hrs reduces the average consumption to 100uA.
In this hypothesis even non-rechargeable batteries, AAA type, will allow a 700mA/100uA = 7000 hrs duration, assuming 700mA/h capacity for the AAA batteries at half power.
Even in this case a duration that suits the purpose is obtained, if palmtop system's durations, quite adequate to a programmed maintenance of 1000 different devices, could be foreseen.