LAMPIRAN A

Listing Program

Program pada Microsoft Visual Basic 6.0

A-1
LISTING PROGRAM PADA VISUAL BASIC 6.0

Dim t0 As Single
Dim dummy As Integer
Dim a, i1, i2, i3, i4, i5, p1, p2, p3, p4, p5, x1, x2, x3 As Integer

Private Sub Command1_Click()
    If p1 = 1 And a < x1 / 2 Then
        i1 = 1
    End If
End Sub

Private Sub Command2_Click()
    If p2 = 1 And a < x2 / 2 Then
        i2 = 1
    End If
End Sub

Private Sub Command3_Click()
    If p3 = 1 And a < x3 / 2 Then
        i3 = 1
    End If
End Sub

Private Sub Command4_Click()
    If i5 = 0 Then
        i4 = 1
    End If
End Sub

Private Sub Command5_Click()
    If i4 = 0 Then
i5 = 1
End If
End Sub

Private Sub Command6_Click()
If p1 = 1 Then
x2 = Val(InputBox("lama hijau jalur 2", "$INPUT' input", 6))
x3 = Val(InputBox("lama hijau jalur 3", "$INPUT' input", 6))
End If

If p2 = 1 Then
x1 = Val(InputBox("lama hijau jalur 1", "$INPUT' input", 6))
x3 = Val(InputBox("lama hijau jalur 3", "$INPUT' input", 6))
End If

If p3 = 1 Then
x1 = Val(InputBox("lama hijau jalur 1", "$INPUT' input", 6))
x2 = Val(InputBox("lama hijau jalur 2", "$INPUT' input", 6))
End If

End Sub

Private Sub Form_Load()
mscomm1.CommPort = 4 'diset sesuai dengan yang ada di device manage
mscomm1.InputMode = comInputModeBinary 'set binary input mode
mscomm1.PortOpen = True 'open the port

IOValA = 0
IOValB = 0
IOValC = 255
x1 = 6  
x2 = 6  
x3 = 6  

mscomm1.Output = "!A" + Chr$(IOValA)  
mscomm1.Output = "!B" + Chr$(IOValB)  
mscomm1.Output = "!C" + Chr$(IOValC)  

i1 = 0  
i2 = 0  
i3 = 0  
i4 = 0  
i5 = 0  
p1 = 0  
p2 = 0  
p3 = 0  
p4 = 0  
p5 = 0  
Timer1.Enabled = True  
Timer2.Enabled = False  
Timer3.Enabled = False  
Picture3.Visible = True  

IOValA = 76  

mscomm1.Output = "A" + Chr$(IOValA)  
Picture2.Visible = False  
Picture1.Visible = False  
Picture4.Visible = True  
Picture5.Visible = False  
Picture6.Visible = False  
Picture7.Visible = True  
Picture8.Visible = False  
Picture9.Visible = False
Private Sub Timer1_Timer()
    p1 = 1
    p2 = 0
    p3 = 0
    p4 = 0
    p5 = 0

    If i1 = 1 Then
        If a > (x1 - 3) And a < x1 Then
            Picture2.Visible = True
            IOValA = 74 ' I/O2 high, all the rest low
            mscomm1.Output = "A" + Chr$(IOValA)
            Picture1.Visible = False
            Picture3.Visible = False
            Picture13.Visible = False
            Picture14.Visible = True
            IOValB = 40
            mscomm1.Output = "B" + Chr$(IOValB)
            Picture15.Visible = False
        End If
    End If
End Sub
ElseIf a > (x1 - 1) And a < (x1 + 2) Then
Picture1.Visible = True
    IOValA = 73
    mscomm1.Output = "A" + Chr$(IOValA)
Picture2.Visible = False
Picture3.Visible = False
Picture13.Visible = True
    IOValB = 24
    mscomm1.Output = "B" + Chr$(IOValB)
Picture14.Visible = False
Picture15.Visible = False
ElseIf a > (x1 + 2) Then
Picture1.Visible = True
    IOValA = 97
    mscomm1.Output = "A" + Chr$(IOValA)
Picture2.Visible = False
Picture3.Visible = False
Picture13.Visible = False
Picture14.Visible = False
Picture15.Visible = True
    IOValB = 72
    mscomm1.Output = "B" + Chr$(IOValB)
Picture6.Visible = True
Picture5.Visible = False
Picture4.Visible = False
Timer2.Enabled = True
Timer1.Enabled = False
a = 0
i1 = 0
End If
ElseIf i4 = 1 Then
If a > (x1 - 3) And a < x1 Then
Picture2.Visible = True
IOValA = 74
    mscomm1.Output = "A" + Chr$(IOValA)
Picture1.Visible = False
Picture3.Visible = False
Picture13.Visible = False
Picture14.Visible = False
Picture15.Visible = True
    IOValB = 72
    mscomm1.Output = "B" + Chr$(IOValB)
ElseIf a > (x1 - 1) And a < (x1 + 2) Then
Picture1.Visible = True
IOValA = 73
    mscomm1.Output = "A" + Chr$(IOValA)
Picture2.Visible = False
Picture3.Visible = False
Picture13.Visible = False
Picture14.Visible = False
Picture15.Visible = True
    IOValB = 72
    mscomm1.Output = "B" + Chr$(IOValB)
If a = (x1 + 1) Then
    If i4 = 1 Then
        Call Pause(2)
    End If
    i1 = 0
    i4 = 0
    i5 = 0
End If
ElseIf a > (x1 + 2) Then
    Picture1.Visible = True
    IOValA = 97
    mscomm1.Output = "A" + Chr$(IOValA)
    Picture2.Visible = False
    Picture3.Visible = False
    Picture13.Visible = False
    Picture14.Visible = False
    Picture15.Visible = True
    IOValB = 72
    mscomm1.Output = "B" + Chr$(IOValB)
    Picture6.Visible = True
    Picture5.Visible = False
    Picture4.Visible = False
    Timer2.Enabled = True
    Timer1.Enabled = False
    a = 0
    i4 = 0
ElseIf i5 = 1 Then
    If a > (x1 - 3) And a < x1 Then
        Picture2.Visible = True
        IOValA = 74
        mscomm1.Output = "A" + Chr$(IOValA)
        Picture1.Visible = False
        Picture3.Visible = False
        Picture13.Visible = False
        Picture14.Visible = True
        IOValB = 40
        mscomm1.Output = "B" + Chr$(IOValB)
ElseIf a > (x1 - 1) And a < (x1 + 2) Then
Picture1.Visible = True
    IOValA = 73
    mscomm1.Output = "A" + Chr$(IOValA)
Picture2.Visible = False
Picture3.Visible = False
Picture13.Visible = True
    IOValB = 24
    mscomm1.Output = "B" + Chr$(IOValB)
Picture14.Visible = False
Picture15.Visible = False
If a = (x1 + 1) Then
    If i5 = 1 Then
        Call Pause(2)
    End If
    i1 = 0
    i4 = 0
    i5 = 0
End If
ElseIf a > (x1 + 2) Then
Picture1.Visible = True
    IOValA = 97
    mscomm1.Output = "A" + Chr$(IOValA)
Picture2.Visible = False
Picture3.Visible = False
Picture13.Visible = False
Picture14.Visible = False
Picture15.Visible = True
    IOValB = 72
    mscomm1.Output = "B" + Chr$(IOValB)
Picture6.Visible = True
ElseIf i1 = 0 Or i4 = 0 Or i5 = 0 Then

    If a > (x1 - 1) And a < (x1 + 2) Then
    Picture2.Visible = True
        IOValA = 74
        mscomm1.Output = "A" + Chr$(IOValA)
    Picture1.Visible = False
    Picture3.Visible = False
    Picture13.Visible = False
    Picture14.Visible = False
    Picture15.Visible = True
        IOValB = 72
        mscomm1.Output = "B" + Chr$(IOValB)
    ElseIf a > (x1 + 2) Then
    Picture1.Visible = True
        IOValA = 97
        mscomm1.Output = "A" + Chr$(IOValA)
    Picture2.Visible = False
    Picture3.Visible = False
    Picture13.Visible = False
    Picture14.Visible = False
    Picture15.Visible = True
        IOValB = 72
        mscomm1.Output = "B" + Chr$(IOValB)
Private Sub Timer2_Timer()
p1 = 0
p2 = 1
p3 = 0
p4 = 0
p5 = 0

If i2 = 1 Then
    If a > (x2 - 3) And a < x2 Then
        Picture5.Visible = True
        IOValA = 81
        mscomm1.Output = "A" + Chr$(IOValA)
        Picture4.Visible = False
        Picture6.Visible = False
        Picture10.Visible = False
        Picture11.Visible = True
        IOValB = 68
        mscomm1.Output = "B" + Chr$(IOValB)
    End If
End If
a = a + 1
End Sub
LookAndFeel = False
ElseIf a > (x2 - 1) And a < (x2 + 2) Then
    Picture4.Visible = True
    IOValA = 73
    mscomm1.Output = "A" + Chr$(IOValA)
    Picture5.Visible = False
    Picture6.Visible = False
    Picture10.Visible = True
    IOValB = 66
    mscomm1.Output = "B" + Chr$(IOValB)
    Picture11.Visible = False
    Picture12.Visible = False

    ElseIf a = (x2 + 2) Then
        Picture4.Visible = True
        IOValA = 9
        mscomm1.Output = "A" + Chr$(IOValA)
        Picture5.Visible = False
        Picture6.Visible = False
        Picture9.Visible = True
        IOValB = 73
        mscomm1.Output = "B" + Chr$(IOValB)
        Picture8.Visible = False
        Picture7.Visible = False
        Timer3.Enabled = True
        Timer2.Enabled = False
        a = 0
        i2 = 0
    End If

ElseIf i4 = 1 Or i5 = 1 Then
    If a > (x2 - 3) And a < x2 Then
ElseIf a > (x2 - 1) And a < (x2 + 2) Then
Picture4.Visible = True
IOValA = 73
mscomm1.Output = "A" + Chr$(IOValA)
Picture5.Visible = False
Picture6.Visible = False
IOValB = 72
mscomm1.Output = "B" + Chr$(IOValB)
ElseIf a > (x2 + 2) Then
Picture4.Visible = True
IOValA = 9
mscomm1.Output = "A" + Chr$(IOValA)
Picture5.Visible = False
Picture6.Visible = False
Picture9.Visible = True
IOValB = 73
End If
If a = (x2 + 1) Then
If i4 = 1 Or i5 = 1 Then
    Call Pause(2)
End If
i2 = 0
i4 = 0
i5 = 0
End If
ElseIf a > (x2 + 2) Then
mscomm1.Output = "B" + Chr$(IOValB)
Picture8.Visible = False
Picture7.Visible = False
Timer3.Enabled = True
Timer2.Enabled = False
a = 0
i4 = 0
i5 = 0
End If

ElseIf i2 = 0 Or i4 = 0 Or i5 = 0 Then
If a > (x2 - 1) And a < (x2 + 2) Then
Picture5.Visible = True
    IOValA = 81
    mscomm1.Output = "A" + Chr$(IOValA)
Picture4.Visible = False
Picture6.Visible = False
ElseIf a > (x2 + 2) Then
Picture4.Visible = True
    IOValA = 9
    mscomm1.Output = "A" + Chr$(IOValA)
Picture5.Visible = False
Picture6.Visible = False
Picture9.Visible = True
    IOValB = 73
    mscomm1.Output = "B" + Chr$(IOValB)
Picture8.Visible = False
Picture7.Visible = False
Timer3.Enabled = True
Timer2.Enabled = False
a = 0
End If
End If
a = a + 1
End Sub

Private Sub Timer3_Timer()
p1 = 0
p2 = 0
p3 = 1
p4 = 0
p5 = 0

If i3 = 1 Then
    If a > (x3 - 3) And a < x3 Then
        Picture8.Visible = True
        IOValA = 137
        mscomm1.Output = "A" + Chr$(IOValA)
        Picture7.Visible = False
        Picture9.Visible = False
        Picture10.Visible = False
        Picture11.Visible = True
        IOValB = 68
        mscomm1.Output = "B" + Chr$(IOValB)
        Picture12.Visible = False
    ElseIf a > (x3 - 1) And a < (x3 + 2) Then
        Picture7.Visible = True
        IOValA = 73
        mscomm1.Output = "A" + Chr$(IOValA)
        Picture8.Visible = False
        Picture9.Visible = False
    End If
End If
Lampiran

Picture10.Visible = True
    IOValB = 66
    mscomm1.Output = "B" + Chr$(IOValB)
Picture11.Visible = False
Picture12.Visible = False
ElseIf a > (x3 + 2) Then
    Picture7.Visible = True
    IOValA = 76
    mscomm1.Output = "A" + Chr$(IOValA)
Picture8.Visible = False
Picture9.Visible = False
Picture10.Visible = False
Picture11.Visible = False
Picture12.Visible = True
    IOValB = 72
    mscomm1.Output = "B" + Chr$(IOValB)
Picture3.Visible = True
Picture2.Visible = False
Picture1.Visible = False
Timer1.Enabled = True
Timer3.Enabled = False
a = 0
i3 = 0
End If

ElseIf i4 = 1 Or i5 = 1 Then
    If a > (x3 - 3) And a < (x3) Then
        Picture8.Visible = True
        IOValA = 137
        mscomm1.Output = "A" + Chr$(IOValA)
Picture7.Visible = False
Picture9.Visible = False

Universitas Kristen Maranatha
Lampiran

Picture10.Visible = False
Picture11.Visible = False
Picture12.Visible = True
IOValB = 72
mscomm1.Output = "B" + Chr$(IOValB)
ElseIf a > (x3 - 1) And a < (x3 + 2) Then
Picture7.Visible = True
IOValA = 137
mscomm1.Output = "A" + Chr$(IOValA)
Picture8.Visible = False
Picture9.Visible = False
Picture11.Visible = False
Picture10.Visible = False
Picture11.Visible = False
Picture12.Visible = True
IOValB = 72
mscomm1.Output = "B" + Chr$(IOValB)
If a = (x3 + 1) Then
If i4 = 1 Or i5 = 1 Then
Call Pause(2)
End If

i3 = 0
i4 = 0
i5 = 0
End If

ElseIf a > (x3 + 2) Then
Picture7.Visible = True
IOValA = 76
mscomm1.Output = "A" + Chr$(IOValA)
Picture8.Visible = False
Picture9.Visible = False
Picture10.Visible = False
Picture11.Visible = False
Picture12.Visible = True
IOValB = 72
    mscomm1.Output = "B" + Chr$(IOValB)
Picture3.Visible = True
Picture2.Visible = False
Picture1.Visible = False
Timer1.Enabled = True
Timer3.Enabled = False
a = 0
i4 = 0
i5 = 0
End If

ElseIf i3 = 0 Or i4 = 0 Or i5 = 0 Then
If a > (x3 - 1) And a < (x3 + 2) Then
Picture8.Visible = True
    IOValA = 137
    mscomm1.Output = "A" + Chr$(IOValA)
Picture7.Visible = False
Picture9.Visible = False
Picture10.Visible = False
Picture11.Visible = False
Picture12.Visible = True
    IOValB = 72
    mscomm1.Output = "B" + Chr$(IOValB)
ElseIf a > (x3 + 2) Then
Picture7.Visible = True
    IOValA = 76
    mscomm1.Output = "A" + Chr$(IOValA)
Picture8.Visible = False
Picture9.Visible = False
Picture10.Visible = False
Picture11.Visible = False
Picture12.Visible = True
IOValB = 72
   mscomm1.Output = "B" + Chr$(IOValB)
Picture3.Visible = True
Picture2.Visible = False
Picture1.Visible = False
Timer1.Enabled = True
Timer3.Enabled = False
a = 0
End If

End If
a = a + 1
End Sub

Sub Pause(ByVal nSecond As Single)
t0 = Timer

    Do While Timer - t0 < nSecond
dummy = DoEvents()
    If Timer < t0 Then
        t0 = t0 - 24 * 60 * 60
    End If
    Loop
End Sub
Private Sub Timer4_Timer()
    mscomm1.Output = "c" ' Request data dari port C
    T = Timer ' Use the Timer to allow the program to continue if there is an error
    While Timer < T + 0.4 And mscomm1.InBufferCount < 1
        Wend
        If 'mscomm1.InBufferCount <> 1 Then
            Call MsgBox("Read Timeout", vbInformation, "USBIO24 VERSION 3 Error")
            Exit Sub
        End If
        Dim TempBuffer As Variant
        Dim ByteBuffer() As Byte
        TempBuffer = mscomm1.Input
        ByteBuffer = TempBuffer
        portCVal = ByteBuffer(0)

        If portCVal = (225 And 31) Then
            If p1 = 1 Then
                i1 = 1
                Text1.Text = "set"
            If a > 3 Then
                i1 = 0
            End If
        End If
        If portCVal = (226 And 31) Then
            If p2 = 1 Then
                i2 = 1
                If a > 3 Then
                    i2 = 0
                End If
            End If
        End If
End If
If portCVal = (228 And 31) Then
    If p3 = 1 Then
        i3 = 1
    If a > 3 Then
        i3 = 0
    End If
End If
End If
If portCVal = (232 And 31) Then
    i4 = 1
End If
If portCVal = (240 And 31) Then
    i5 = 1
End If

If i1 = 1 Then
    Text1.Text = "set1"
ElseIf i2 = 1 Then
    Text1.Text = "set2"
ElseIf i3 = 1 Then
    Text1.Text = "set3"
ElseIf i4 = 1 Then
    Text1.Text = "set4"
ElseIf i5 = 1 Then
    Text1.Text = "set5"
Else
    Text1.Text = "none"
End If
End Sub

Private Sub Timer5_Timer()
    Text2.Text = a
    Text3.Text = x1
    Text4.Text = x2
    Text5.Text = x3
End Sub
LAMPIRAN B

DOKUMENTASI B-1
### Lampiran

#### PENGUJIAN 1

<table>
<thead>
<tr>
<th>Jalur/set awal</th>
<th>Penekanan</th>
<th>Merah/kuning/hijau (detik)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 / 3</td>
<td>0</td>
<td>22/2/1</td>
</tr>
<tr>
<td>2 / default</td>
<td>0</td>
<td>14/3/6</td>
</tr>
<tr>
<td>3 / default</td>
<td>0</td>
<td>14/3/6</td>
</tr>
<tr>
<td>4</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

#### PENGUJIAN 2

<table>
<thead>
<tr>
<th>Jalur/set awal</th>
<th>Penekanan</th>
<th>Merah/kuning/hijau (detik)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 / 4</td>
<td>0</td>
<td>22/2/2</td>
</tr>
<tr>
<td>2 / default</td>
<td>0</td>
<td>15/3/6</td>
</tr>
<tr>
<td>3 / default</td>
<td>0</td>
<td>15/3/6</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>√</td>
<td></td>
</tr>
</tbody>
</table>

#### PENGUJIAN 3

<table>
<thead>
<tr>
<th>Jalur/set awal</th>
<th>Penekanan</th>
<th>Merah/kuning/hijau (detik)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 / default</td>
<td>0</td>
<td>22/2/4</td>
</tr>
<tr>
<td>2 / default</td>
<td>0</td>
<td>17/3/6</td>
</tr>
<tr>
<td>3 / default</td>
<td>0</td>
<td>17/3/6</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>√</td>
<td></td>
</tr>
</tbody>
</table>
Lampiran

<table>
<thead>
<tr>
<th>PENGUJIAN 4</th>
<th>Jalur/set awal</th>
<th>Penekanan</th>
<th>Merah/kuning/hijau (detik)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 / 10</td>
<td>0</td>
<td>22/2/8</td>
<td></td>
</tr>
<tr>
<td>2 / default</td>
<td>0</td>
<td>21/3/6</td>
<td></td>
</tr>
<tr>
<td>3 / default</td>
<td>0</td>
<td>21/3/6</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>√</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PENGUJIAN 5</th>
<th>Jalur/set awal</th>
<th>Penekanan</th>
<th>Merah/kuning/hijau (detik)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 / default</td>
<td>√</td>
<td>20/2/4</td>
<td></td>
</tr>
<tr>
<td>2 / default</td>
<td>0</td>
<td>15/3/6</td>
<td></td>
</tr>
<tr>
<td>3 / default</td>
<td>0</td>
<td>15/3/6</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PENGUJIAN 6</th>
<th>Jalur/set awal</th>
<th>Penekanan</th>
<th>Merah/kuning/hijau (detik)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 / default</td>
<td>0</td>
<td>22/2/4</td>
<td></td>
</tr>
<tr>
<td>2 / default</td>
<td>0</td>
<td>17/3/6</td>
<td></td>
</tr>
<tr>
<td>3 / default</td>
<td>0</td>
<td>17/3/6</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>√</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Lampiran

#### PENGUJIAN 7

<table>
<thead>
<tr>
<th>Jalur/set awal</th>
<th>Penekanan</th>
<th>Merah/kuning/hijau (detik)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 / default</td>
<td>0</td>
<td>18/3/6</td>
</tr>
<tr>
<td>2 / default</td>
<td>√</td>
<td>18/3/6</td>
</tr>
<tr>
<td>3 / default</td>
<td>0</td>
<td>18/3/6</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### PENGUJIAN 8

<table>
<thead>
<tr>
<th>Jalur/set awal</th>
<th>Penekanan</th>
<th>Merah/kuning/hijau (detik)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 / default</td>
<td>0</td>
<td>14/3/6</td>
</tr>
<tr>
<td>2 / 3</td>
<td>0</td>
<td>22/2/1</td>
</tr>
<tr>
<td>3 / default</td>
<td>0</td>
<td>14/3/6</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>√</td>
<td></td>
</tr>
</tbody>
</table>

#### PENGUJIAN 9

<table>
<thead>
<tr>
<th>Jalur/set awal</th>
<th>Penekanan</th>
<th>Merah/kuning/hijau (detik)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 / default</td>
<td>0</td>
<td>19/3/6</td>
</tr>
<tr>
<td>2 / 10</td>
<td>√</td>
<td>20/2/8</td>
</tr>
<tr>
<td>3 / default</td>
<td>0</td>
<td>19/3/6</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Lampiran

**PENGUJIAN 10**

<table>
<thead>
<tr>
<th>Jalur/set awal</th>
<th>Penekanan</th>
<th>Merah/kuning/hijau (detik)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 / default</td>
<td>0</td>
<td>21/3/6</td>
</tr>
<tr>
<td>2 / 10</td>
<td>0</td>
<td>22/2/8</td>
</tr>
<tr>
<td>3 / default</td>
<td>0</td>
<td>21/3/6</td>
</tr>
<tr>
<td>4</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

**PENGUJIAN 11**

<table>
<thead>
<tr>
<th>Jalur/set awal</th>
<th>Penekanan</th>
<th>Merah/kuning/hijau (detik)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 / default</td>
<td>0</td>
<td>15/3/6</td>
</tr>
<tr>
<td>2 / default</td>
<td>√</td>
<td>20/2/4</td>
</tr>
<tr>
<td>3 / default</td>
<td>0</td>
<td>15/3/6</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

**PENGUJIAN 12**

<table>
<thead>
<tr>
<th>Jalur/set awal</th>
<th>Penekanan</th>
<th>Merah/kuning/hijau (detik)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 / default</td>
<td>0</td>
<td>17/3/6</td>
</tr>
<tr>
<td>2 / default</td>
<td>0</td>
<td>22/2/4</td>
</tr>
<tr>
<td>3 / default</td>
<td>0</td>
<td>17/3/6</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>√</td>
<td></td>
</tr>
</tbody>
</table>
### PENGUJIAN 13

<table>
<thead>
<tr>
<th>Jalur/set awal</th>
<th>Penekanan</th>
<th>Merah/kuning/hijau (detik)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 / default</td>
<td>0</td>
<td>18/3/6</td>
</tr>
<tr>
<td>2 / default</td>
<td>0</td>
<td>18/3/6</td>
</tr>
<tr>
<td>3 / default</td>
<td>√</td>
<td>18/3/6</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

### PENGUJIAN 14

<table>
<thead>
<tr>
<th>Jalur/set awal</th>
<th>Penekanan</th>
<th>Merah/kuning/hijau (detik)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 / default</td>
<td>0</td>
<td>14/3/6</td>
</tr>
<tr>
<td>2 / default</td>
<td>0</td>
<td>14/3/6</td>
</tr>
<tr>
<td>3 / default</td>
<td>0</td>
<td>22/2/1</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>√</td>
<td></td>
</tr>
</tbody>
</table>

### PENGUJIAN 15

<table>
<thead>
<tr>
<th>Jalur/set awal</th>
<th>Penekanan</th>
<th>Merah/kuning/hijau (detik)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 / default</td>
<td>0</td>
<td>17/3/6</td>
</tr>
<tr>
<td>2 / default</td>
<td>0</td>
<td>17/3/6</td>
</tr>
<tr>
<td>3 / default</td>
<td>0</td>
<td>22/2/4</td>
</tr>
<tr>
<td>4</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>
**Lampiran**

### PENGUJIAN 16

<table>
<thead>
<tr>
<th>Jalur/set awal</th>
<th>Penekanan</th>
<th>Merah/kuning/hijau (detik)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 / default</td>
<td>0</td>
<td>19/3/6</td>
</tr>
<tr>
<td>2 / default</td>
<td>0</td>
<td>19/3/6</td>
</tr>
<tr>
<td>3 / 10</td>
<td>√</td>
<td>20/2/8</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

### PENGUJIAN 17

<table>
<thead>
<tr>
<th>Jalur/set awal</th>
<th>Penekanan</th>
<th>Merah/kuning/hijau (detik)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 / default</td>
<td>0</td>
<td>26/3/6</td>
</tr>
<tr>
<td>2 / default</td>
<td>0</td>
<td>26/3/6</td>
</tr>
<tr>
<td>3 / 15</td>
<td>0</td>
<td>22/2/13</td>
</tr>
<tr>
<td>4</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

### PENGUJIAN 18

<table>
<thead>
<tr>
<th>Jalur/set awal</th>
<th>Penekanan</th>
<th>Merah/kuning/hijau (detik)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 / default</td>
<td>0</td>
<td>15/3/6</td>
</tr>
<tr>
<td>2 / default</td>
<td>0</td>
<td>15/3/6</td>
</tr>
<tr>
<td>3 / default</td>
<td>√</td>
<td>20/2/4</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

---

*Universitas Kristen Maranatha*
PENGUJIAN 19

<table>
<thead>
<tr>
<th>Jalur/set awal</th>
<th>Penekanan</th>
<th>Merah/kuning/hijau (detik)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 / default</td>
<td>0</td>
<td>17/3/6</td>
</tr>
<tr>
<td>2 / default</td>
<td>0</td>
<td>17/3/6</td>
</tr>
<tr>
<td>3 / default</td>
<td>0</td>
<td>22/2/6</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>√</td>
<td></td>
</tr>
</tbody>
</table>

PENGUJIAN 20

<table>
<thead>
<tr>
<th>Jalur/set awal</th>
<th>Penekanan</th>
<th>Merah/kuning/hijau (detik)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 / default</td>
<td>√</td>
<td>18/3/6</td>
</tr>
<tr>
<td>2 / default</td>
<td>0</td>
<td>18/3/6</td>
</tr>
<tr>
<td>3 / default</td>
<td>0</td>
<td>18/3/6</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>
LAMPIRAN C

Datasheet USBIO24DIP C-1
Datasheet FT245BL C-6
Datasheet SX52BD C-10
USB I/O 24 Version 3 Datasheet

Module Standard Firmware & Software

- Virtual Comm Port driver allows access as a regular serial port.
- Optional DLL based driver available.
- Easy to program from popular languages C, Basic, Delphi, etc.
- Simple command set for easy control of ports and data transfer.

Module Layout and Physical Dimensions

Port Connector Pinout

USBIO24 VERSION 3 Digital I/O Module

The USBIO24 VERSION 3 is the second generation of a low-cost integrated module for the input and/or output of digital signals from a computer system by connection to the USB port. The module pin out and firmware are 100% compatible with the first version of the USB I/O24.

The module features 24 5V level signal lines individually programmable as input or output as well as capabilities for further expansion via the expansion port. As the module connects to the USB port, multiple modules can be connected to a single PC by the use of a USB hub or hubs. Each module features a serial number and the PC can identify each module uniquely allowing for multiple modules to be connected for a single application. The outputs of the module are able to source or sink up to 30mA per I/O to allow for direct connection to a variety of devices.

The USB I/O 24 Version 3 Module

MODULE FEATURES

- 24 independently programmable Input / Output Pins Grouped into 3 ports.
- Single module High-Speed Digital Input / Output solution.
- Up to 128 modules can be connected to a single PC with capabilities of further expansion.
- Easy to connect by 0.1" pitch headers to suit standard IDC connectors.
- Integrated Type-B USB Connector.
- On-board unique serial number in EEPROM and custom programmable FLASH microcontroller.
- Both USB Enumeration information and Microcontroller can be re-programmed to suit customer needs.
- Module powered by the USB from the PC or via an external power source.
**USB I/O 24 Version 3 Datasheet**

**Communication Protocol (Standard Firmware Only)**

<table>
<thead>
<tr>
<th>COMMAND</th>
<th>DATA</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>'?'</td>
<td></td>
<td>Identify Device</td>
</tr>
<tr>
<td>'A'</td>
<td>1 Byte Port Data</td>
<td>Write to Port A</td>
</tr>
<tr>
<td>'B'</td>
<td>1 Byte Port Data</td>
<td>Write to Port B</td>
</tr>
<tr>
<td>'C'</td>
<td>1 Byte Port Data</td>
<td>Write to Port C</td>
</tr>
<tr>
<td>'a'</td>
<td>Responds with 1 Byte Port Data</td>
<td>Read from Port A</td>
</tr>
<tr>
<td>'b'</td>
<td>Responds with 1 Byte Port Data</td>
<td>Read from Port B</td>
</tr>
<tr>
<td>'c'</td>
<td>Responds with 1 Byte Port Data</td>
<td>Read from Port C</td>
</tr>
<tr>
<td>'A'</td>
<td>1 Byte Port I/O Data</td>
<td>Write to Port A Direction Register</td>
</tr>
<tr>
<td>'B'</td>
<td>1 Byte Port I/O Data</td>
<td>Write to Port B Direction Register</td>
</tr>
<tr>
<td>'C'</td>
<td>1 Byte Port I/O Data</td>
<td>Write to Port C Direction Register</td>
</tr>
</tbody>
</table>

The commands in the above table are in ASCII format. All Data is sent in Binary format.

**Communication Protocol (VERSION 3 Standard Firmware Only)**

**New Features of VERSION 3 Firmware**
- Transmits on Pin Change without Software Polling (Mode 2)
- Enables setting of the pin pull up feature on SNS2
- Enables setting of inputs as CMOS Level or TTL Level or Schmitt Trigger Inputs

<table>
<thead>
<tr>
<th>COMMAND</th>
<th>DATA</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>'?'</td>
<td></td>
<td>Identify Device</td>
</tr>
<tr>
<td>'A'</td>
<td>1 Byte Port Data</td>
<td>Write to Port A</td>
</tr>
<tr>
<td>'B'</td>
<td>1 Byte Port Data</td>
<td>Write to Port B</td>
</tr>
<tr>
<td>'C'</td>
<td>1 Byte Port Data</td>
<td>Write to Port C</td>
</tr>
<tr>
<td>'a'</td>
<td>Responds with 1 Byte Port Data</td>
<td>Read from Port A</td>
</tr>
<tr>
<td>'b'</td>
<td>Responds with 1 Byte Port Data</td>
<td>Read from Port B</td>
</tr>
<tr>
<td>'c'</td>
<td>Responds with 1 Byte Port Data</td>
<td>Read from Port C</td>
</tr>
<tr>
<td>'A'</td>
<td>1 Byte Port I/O Data</td>
<td>Write to Port A Direction Register</td>
</tr>
<tr>
<td>'B'</td>
<td>1 Byte Port I/O Data</td>
<td>Write to Port B Direction Register</td>
</tr>
<tr>
<td>'C'</td>
<td>1 Byte Port I/O Data</td>
<td>Write to Port C Direction Register</td>
</tr>
<tr>
<td>'α'</td>
<td>Follow with Port write and binary data</td>
<td>Port Pull up feature for SNS2</td>
</tr>
<tr>
<td>'β'</td>
<td>Follow with Port write and binary data</td>
<td>Set inputs to CMOS/TTL level</td>
</tr>
<tr>
<td>'σ'</td>
<td>Follow with Port write and binary data</td>
<td>Set Port Schmitt trigger enables</td>
</tr>
<tr>
<td>'2'</td>
<td></td>
<td>Sets unit to Mode 2 (Enables transmit pin change)</td>
</tr>
<tr>
<td>'1'</td>
<td></td>
<td>Sets unit to Mode 1 (Exits Mode 2)</td>
</tr>
</tbody>
</table>

The commands in the above table are in ASCII format. All Data is sent in Binary format.

---

**USB I/O 24 I/O Connector (PORT A, B, C)**

**All 3 ports**

<table>
<thead>
<tr>
<th>PIN #</th>
<th>SIGNAL</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>VCC</td>
<td>PWR</td>
<td>+5V from USB BUS – May be used to supply your circuitry up to the maximum of 50mA</td>
</tr>
<tr>
<td>2</td>
<td>I/O</td>
<td>I/O</td>
<td>Programmable I/O pin with bit value of 128</td>
</tr>
<tr>
<td>3</td>
<td>I/O</td>
<td>I/O</td>
<td>Programmable I/O pin with bit value of 64</td>
</tr>
<tr>
<td>4</td>
<td>I/O</td>
<td>I/O</td>
<td>Programmable I/O pin with bit value of 32</td>
</tr>
<tr>
<td>5</td>
<td>I/O</td>
<td>I/O</td>
<td>Programmable I/O pin with bit value of 16</td>
</tr>
<tr>
<td>6</td>
<td>I/O</td>
<td>I/O</td>
<td>Programmable I/O pin with bit value of 8</td>
</tr>
<tr>
<td>7</td>
<td>I/O</td>
<td>I/O</td>
<td>Programmable I/O pin with bit value of 4</td>
</tr>
<tr>
<td>8</td>
<td>I/O</td>
<td>I/O</td>
<td>Programmable I/O pin with bit value of 2</td>
</tr>
<tr>
<td>9</td>
<td>I/O</td>
<td>I/O</td>
<td>Programmable I/O pin with bit value of 1</td>
</tr>
<tr>
<td>10</td>
<td>GND</td>
<td>PWR</td>
<td>Ground signal USB BUS and all I/O</td>
</tr>
</tbody>
</table>

**USB I/O 24 Port Expansion Connector (PORT EXP)**

<table>
<thead>
<tr>
<th>PIN #</th>
<th>SIGNAL</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>VCC</td>
<td>PWR</td>
<td>+5V from USB BUS – May be used to supply your circuitry up to the maximum of 50mA</td>
</tr>
<tr>
<td>2</td>
<td>D7</td>
<td>I/O</td>
<td>I/O pin with bit value of 128</td>
</tr>
<tr>
<td>3</td>
<td>D6</td>
<td>I/O</td>
<td>I/O pin with bit value of 64</td>
</tr>
<tr>
<td>4</td>
<td>D5</td>
<td>I/O</td>
<td>I/O pin with bit value of 32</td>
</tr>
<tr>
<td>5</td>
<td>D4</td>
<td>I/O</td>
<td>I/O pin with bit value of 16</td>
</tr>
<tr>
<td>6</td>
<td>D3</td>
<td>I/O</td>
<td>I/O pin with bit value of 8</td>
</tr>
<tr>
<td>7</td>
<td>D2</td>
<td>I/O</td>
<td>I/O pin with bit value of 4</td>
</tr>
<tr>
<td>8</td>
<td>D1</td>
<td>I/O</td>
<td>I/O pin with bit value of 2</td>
</tr>
<tr>
<td>9</td>
<td>D0</td>
<td>I/O</td>
<td>I/O pin with bit value of 1</td>
</tr>
<tr>
<td>10</td>
<td>A0</td>
<td>I/O</td>
<td>RA4 on SNS2</td>
</tr>
<tr>
<td>11</td>
<td>A1</td>
<td>I/O</td>
<td>RA5 on SNS2</td>
</tr>
<tr>
<td>12</td>
<td>A2</td>
<td>I/O</td>
<td>RA6 on SNS2</td>
</tr>
<tr>
<td>13</td>
<td>A3</td>
<td>I/O</td>
<td>RA7 on SNS2</td>
</tr>
<tr>
<td>14</td>
<td>GND</td>
<td>PWR</td>
<td>Ground signal USB BUS and all I/O</td>
</tr>
</tbody>
</table>
Programmers Reference Documentation

Programming the USBIO24 VERSION 3 from Visual Basic using the Virtual Comm Port.

To operate the USBIO24 VERSION 3 from within Visual Basic it’s best to use the Microsoft MSComm control to access the com port. To input data from the USBIO24 VERSION 3 you must use the port in binary mode and receiving the data is a bit convoluted.

Opening the Port

As the USBIO24 VERSION 3 unit uses binary data transfer we must use the port in binary mode.
If the port number is incorrect or the USBIO24 VERSION 3 module is not connected then VB will generate an error.

MSComm1.ComPort = 3 ' Set this number as shown in the Device Manager
MSComm1.InputMode = vbInputModeBinary ' Set Binary Input Mode
MSComm1.PortOpen = True ' Open the Port

Setting the pins as Inputs or Outputs

Setting the Ports as Input or Output you must determine the value for the pins you want set as inputs.
To set pins I/O1, I/O2 and I/O3 as inputs and the remaining pins as outputs you simply add the bit values of the input pins 1 + 2 + 4 = 7 and thus the value to be placed in the IOValx variable in the following example code is 7.

' Set I/O1, I/O2 & I/O3 of port A to inputs and the rest as outputs.
IOValA = 7 ' First 3 inputs all the rest as outputs
IOValB = 0 ' All outputs
IOValC = 0 ' All outputs
MSComm1.Output = "BA" + Chr$(IOValA) ' Write to Port A Direction Register
MSComm1.Output = "BB" + Chr$(IOValB) ' Write to Port B Direction Register
MSComm1.Output = "CC" + Chr$(IOValC) ' Write to Port C Direction Register

Writing to the Ports

To write to the Output Pins simple repeat the above without the ! character.
The following example code sets the I/O8 pin on port B to high and the remaining pins as low.

IOValB = 128 ' I/O8 high, all the rest low
MSComm1.Output = "AB" + Chr$(IOValB)

Mode 2 Functional Changes

All reads have a port designator ("a", "b" or "c") before the data.
All auto sends have a port designator ("a", "b" or "c") before the data.
All writes to the port that change the port will result in a port data auto send.

Driver Installation

Your first choice when using the USBIO24 VERSION 3 is whether you want to use the Virtual COM Port driver or the Direct DLL driver.

For programming simplicity the best driver is the Virtual COM Port and when installed the USBIO24 VERSION 3 will appear in the System Properties / Device Manager as an USB Serial Port (COMXn) as follows.

The COM port number will vary depending on the number of existing COM ports on your computer and the number of USBIO24 VERSION 3s or USBSMODs connected to your system.

To install the Virtual COM Port drivers, download the driver from our website or the f1chip.com website and unzip the files to a local directory. Then connect the USBIO24 VERSION 3 and windows will automatically ask for the driver, select to specify a location and browse to the directory where you have unzipped the files. (Use of the Non Plug & Play driver for the USBIO24 VERSION 3 is recommended to avoid a delay in connecting the USBIO24 VERSION 3)

Once the Virtual COM Port is installed it can be programmed exactly as a regular serial COM port using the MSComm control from within VB or API calls from C or other languages. Set the COM port to the same number as appears in the Device Manager, the baud rate, stop bits, parity etc are not used as the device always runs at full speed.

The Direct DLL driver is installed in a similar manner but using the alternative download from the website.

Programming the Direct DLL driver is by call to the DLL Library functions.
Lampiran

Universitas Kristen Maranatha
Lampiran

Universitas Kristen Maranatha

1.1 General Description

The FT245BL provides an easy cost-effective method of transferring data to/from a peripheral and a host PC at up to 8.3 Million bits (1 Megabyte) per second. Its simple, FIFO-like design makes it easy to interface to any microcontroller or microprocessor via I/O ports.

To send data from the peripheral to the host computer, simply write the byte-wide data into the module when TXE# is low. If the (34-byte) transmit buffer fills up or if busy storing the previously written byte, the device keeps TXE# high in order to stop further data from being written until some of the FIFO data has been transferred over USB to the host. TXE# goes high after every byte written.

When the host sends data to the peripheral over USB, the device will take RXF# low to let the peripheral know that at least one byte of data is available. The peripheral can read a data byte every time RXF# goes low: RXF# goes high after every byte read.

By using FTDI's virtual COM port drivers, the peripheral looks like a standard COM port to the application software. Commands to set the baud rate are ignored - the device always transfers data at its fastest rate regardless of the application's baud-rate setting. Alternatively, FTDI's DG8X drivers allow application software to access the device 'directly' through a published DLL based API. Details of the current VCP and DG8X driver can be found on FTDI's web site [http://www.ftdichip.com/].

2.0 Enhancements

This section summarises the enhancements of the second generation device compared to its FT82245AM predecessor. For further details, consult the device pin-out description and functional descriptions.

- **Integrated Power-On-Reset (POR) Circuit**
  - The device now incorporates an internal POR function. The existing RESET# pin is maintained in order to allow external logic to reset the device when required; however, for many applications this pin can now be left high or hard wired to VCC. In addition, a new reset output pin (RSTOUT) is provided in order to allow the new POR circuit to provide a stable reset to external MCU and other devices. RSTOUT was the TEST pin on the previous generation of devices.

- **Integrated Level Converter on FIFO interface and control signals**
  - The previous devices would drive the FIFO and control signals at 5V CMOS logic levels. The new device has a separate VCCIO pin allowing the device to directly interface to 3.3V and other logic families without the need for external level converter IC's.

- **Power Management control for USB Bus**
  - A new PWREN# signal is provided which can be used to directly drive a transistor or P-Channel MOSFET in applications where power switching of external circuitry is required. A new EEPROM based option makes the device pull gently down its FIFO interface lines when the power is shut off (PWREN# is high). In this mode, any residual
3.0 Block Diagram (simplified)

3.1 Functional Block Descriptions

- **3.3V LDO Regulator**
  The 3.3V LDO Regulator generates the 3.3 volt reference voltage for driving the USB transceiver cell output buffers. It requires an external decoupling capacitor to be attached to the 3V3OUT regulator output pin. It also provides 3.3V power to the RSTOUT pin. The main function of this block is to power the USB Transceiver and the Reset Generator Cells rather than power external logic. However, external circuitry requiring 3.3V nominal at a current of not greater than 5mA could also draw its power from the 3V3OUT pin if required.

- **USB Transceiver**
  The USB Transceiver Cell provides the USB 1.1 / USB 2.0 full-speed physical interface to the USB cable. The output drivers provide 3.3 volt level slew rate control signalling, whilst a differential receiver and two single ended receivers provide USB data in, ESD and USB Reset condition detection.

- **USB DPLL**
  The USB DPLL cell locks on to the incoming NRZI USB data and provides separate recovered clock and data signals to the SIE block.

- **6MHz Oscillator**
  The 6MHz Oscillator generates a 6MHz reference clock for the x8 Clock multiplicator from an external 6MHz crystal or ceramic resonator.

- **x8 Clock Multiplicator**
  The x8 Clock Multiplicator takes the 6MHz input from the Oscillator cell and generates a 12MHz reference clock for the SIE, USB Protocol Engine and FIFO controller blocks. It also generates a 48MHz reference clock for the USB DPLL.

- **EEPROM Interface**
  The EEPROM Interface is used to writes data to the EEPROM. The EEPROM is a non-volatile memory device used to store configuration data. The EEPROM Interface provides the necessary control signals to the EEPROM.

- **FIFO Controller**
  The FIFO Controller manages the USB data buffer between the USB DPLL and the USB transceiver. It also manages the handshake signals between the USB transceiver and the USB host. The FIFO Controller provides a buffer to absorb any difference in data rate between the USB transceiver and the USB host.

- **USB - Parallel I.C.**
  Multiple Device Support without EEPROM
  When no EEPROM (or a blank or invalid EEPROM) is attached to the device, the FT245BL no longer gives a serial number as part of its USB descriptor. This allows multiple devices to be simultaneously connected to the same PC. However, we still highly recommend that EEPROM is used, as without serial numbers a device can only be identified by which hub port in the USB tree it is connected to which can change if the end user re-plugs the device into a different USB port.

- **EEREQ#/EEEN#/ Leafs**
  These FT8U245AM pins are no longer supported on the FT245BL device. They have been replaced with the new SI/WU and PA/RN signals respectively.

- **USB 2.0 (full speed option)**
  A new EEPROM based option: allows the FT245BL to return a USB 2.0 device descriptor as opposed to USB 1.1. Note: The device would be a USB 2.0 Full Speed device (12Mbps) as opposed to a USB 2.0 High Speed device (480Mbps).
4.0 Device Pin-Out

**Figure 1**
Pin-Out
(Lead free LQFP-32 Package)

- **FT245BL USB FIFO (USB - Parallel) I.C.**

- **Serial Interface Engine (SIE)**
The Serial Interface Engine (SIE) block performs the Parallel to Serial and Serial to Parallel conversion of the USB data. In accordance to the USB 2.0 specification, it performs bit stuffing / unstuffing and CRC5 / CRC16 generation / checking on the USB data stream.

- **USB Protocol Engine**
The USB Protocol Engine manages the data stream from the device USB control endpoint. It handles the low level USB protocol (Chapter 9) requests generated by the USB Host controller and the commands for controlling the functional parameters of the FIFO.

- **FIFO Receive Buffer (128 bytes)**
Data sent from the USB Host to the FIFO via the USB data out endpoint is stored in the FIFO Receive Buffer and is removed from the buffer by reading the FIFO contents using RD#.  

- **FIFO Transmit Buffer (384 bytes)**
Data written into the FIFO using WR# is stored in the FIFO Transmit Buffer. The Host removes Data from the FIFO Transmit Data by sending a USB request for data from the device data in endpoint.

- **FIFO Controller**
The FIFO Controller handles the transfer of data between the external FIFO interface pins and the FIFO Transmit and Receive buffers.

- **RESET Generator**
The Reset Generator Cell provides a reliable power-on reset to the device internal circuitry on power up. An additional RESET# input and RSTCUT# output are provided to allow other devices to reset the FT245BL, or the FT245BL to reset other devices respectively. During reset, RSTCUT# is driven low, otherwise it drives out at the 3.3V provided by the onboard regulator. RSTCUT# can be used to control the 1.5K pull-up on USB# directly or delayed USB

**Figure 2**
Pin-Out
(Schematic Symbol)

- **FT245BL USB FIFO (USB - Parallel) I.C.**

  enumeration is required. RSTCUT# will be low for approximately 5ms after VCC has risen above 3.5V and the device oscillator is running. AND RESET# is high. RESET# should be tied to VCC unless it is a requirement to reset the device from external logic or an external reset generator I.C.

- **EEPROM Interface**
  Though the FT245BL will work without the optional EEPROM, an external 93C45 (93C56 or 93C06) EEPROM can be used to customise the USB VID, PID, Serial Number, Product Description Strings and Power Descriptor value of the FT245BL for OEM applications. Other parameters controlled by the EEPROM include Remote Wake Up, Synchronous Transfer Mode, Soft Boot Down on Power-Off and USB 2.0 descriptor modes. The EEPROM should be a 16 bit wide configuration such as a MicroChip 93LC46B or equivalent capable of 1MB clock rate at VCC = 4.35V to 5.25V. The EEPROM is programmable on board over USB using a utility available from FTDI's web site (http://www.ftdichip.com/). This allows a blank part to be soldered onto the PCB and programmed as part of the manufacturing and test process. If no EEPROM is connected (or the EEPROM is blank), the FT245BL will use its built-in default VID, PID Product Description and Power Descriptor Value. In this case, the device will not have a serial number as part of the USB descriptor.
### POWER CONTROL GROUP

<table>
<thead>
<tr>
<th>Pin#</th>
<th>Signal</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>PWREN#</td>
<td>OUT</td>
<td>Goes Low after the device is configured via USB, then high during USB suspend. Can be used to control power to external logic using a P-Channel Logic Level MOSFET switch. Enable the Interface Pull-Down Option in EEPROM when using the PWREN# pin in this way.</td>
</tr>
<tr>
<td>11</td>
<td>SI / WU</td>
<td>IN</td>
<td>The Send Immediate / Wake-Up signal combines two functions on a single pin. If USB is in suspend mode (PWREN# = 1) and remote wake-up is enabled in the EEPROM, strobing this pin low will cause the device to request a resume on the USB Bus. Normally, this can be used to wake up the Host PC. During normal operation (PWREN# = 0), if this pin is strobed low any data in the device Tx buffer will be sent out over USB on the nextBulk-in request from the driver regardless of the pending packet size. This can be used to optimize USB transfer speed for some applications. Tie this pin to VCCO if not used.</td>
</tr>
</tbody>
</table>

### MISCELLANEOUS SIGNAL GROUP

<table>
<thead>
<tr>
<th>Pin#</th>
<th>Signal</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>RESET#</td>
<td>IN</td>
<td>Can be used by an external device to reset the FT245BL. If not required, tie to VCC.</td>
</tr>
<tr>
<td>5</td>
<td>RSTOUT#</td>
<td>OUT</td>
<td>Output of the internal Reset Generator. Stays high impedance for ~5ms after VCC &gt; 3.5V and the internal clock starts up, then changes its output to the 3.3V output of the internal regulator. Taking RESET# low will also force RSTOUT# to drive low. RSTOUT# is NOT affected by a USB Bus Reset.</td>
</tr>
<tr>
<td>27</td>
<td>XTIN</td>
<td>IN</td>
<td>Input to 5MHz Crystal Oscillator. This pin can also be driven by an external 5MHz clock if required. Note: Switching threshold of this pin is VCC/2, so if driving from an external source, the source must be at 5V CMOS level or higher, coupled to centre around VCC/2.</td>
</tr>
<tr>
<td>28</td>
<td>XOUT</td>
<td>OUT</td>
<td>Output from 5MHz Crystal Oscillator. XOUT stops oscillating during USB suspend, so take care if using this signal to clock external logic.</td>
</tr>
<tr>
<td>3F</td>
<td>TEST</td>
<td>IN</td>
<td>Pulls device in I.C. test mode – must be tied to GND for normal operation.</td>
</tr>
</tbody>
</table>

### 4.1 Signal Descriptions

#### Table 1 - FT245BL - PINOUT DESCRIPTION

**FIFO DATA BUS GROUP (*** Note 1)**

<table>
<thead>
<tr>
<th>Pin#</th>
<th>Signal</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>D0</td>
<td>I/O</td>
<td>FIFO Data Bus Bit 0</td>
</tr>
<tr>
<td>24</td>
<td>D1</td>
<td>I/O</td>
<td>FIFO Data Bus Bit 1</td>
</tr>
<tr>
<td>23</td>
<td>D2</td>
<td>I/O</td>
<td>FIFO Data Bus Bit 2</td>
</tr>
<tr>
<td>22</td>
<td>D3</td>
<td>I/O</td>
<td>FIFO Data Bus Bit 3</td>
</tr>
<tr>
<td>21</td>
<td>D4</td>
<td>I/O</td>
<td>FIFO Data Bus Bit 4</td>
</tr>
<tr>
<td>20</td>
<td>D5</td>
<td>I/O</td>
<td>FIFO Data Bus Bit 5</td>
</tr>
<tr>
<td>19</td>
<td>D6</td>
<td>I/O</td>
<td>FIFO Data Bus Bit 6</td>
</tr>
<tr>
<td>18</td>
<td>D7</td>
<td>I/O</td>
<td>FIFO Data Bus Bit 7</td>
</tr>
</tbody>
</table>

**FIFO CONTROL INTERFACE GROUP**

<table>
<thead>
<tr>
<th>Pin#</th>
<th>Signal</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>RD#</td>
<td>IN</td>
<td>Enables Current FIFO Data Byte on D0-D7 when low. Fetches the next FIFO Data Byte (if available) from the Receive FIFO Buffer when RD# goes from low to high. *** Note 1</td>
</tr>
<tr>
<td>15</td>
<td>WR</td>
<td>IN</td>
<td>Writes the Data Byte on the D0-D7 into the Transmit FIFO Buffer when WR goes from high to low. *** Note 1</td>
</tr>
<tr>
<td>14</td>
<td>TXE#</td>
<td>OUT</td>
<td>When high, do not write data into the FIFO. When low, data can be written into the FIFO by strobing WR high then low. *** Note 2</td>
</tr>
<tr>
<td>12</td>
<td>RXF#</td>
<td>OUT</td>
<td>When high, do not read data from the FIFO. When low, there is data available in the FIFO which can be read by strobing RD# low then high again. *** Note 2</td>
</tr>
</tbody>
</table>

**USB INTERFACE GROUP**

<table>
<thead>
<tr>
<th>Pin#</th>
<th>Signal</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>USBCP</td>
<td>I/O</td>
<td>USB Data Signal Plus (Requires 1.5k pull-up to 3V3OUT or RSTOUT#)</td>
</tr>
<tr>
<td>8</td>
<td>USBCM</td>
<td>I/O</td>
<td>USB Data Signal Minus</td>
</tr>
</tbody>
</table>

**EEPROM INTERFACE GROUP**

<table>
<thead>
<tr>
<th>Pin#</th>
<th>Signal</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>EECS</td>
<td>I/O</td>
<td>EEPROM – Chip Select. For 48MHz operation pull EECS to GND using a 10K resistor. For 6MHz operation no resistor is required. *** Note 3</td>
</tr>
<tr>
<td>1</td>
<td>EESK</td>
<td>OUT</td>
<td>Clock signal to EEPROM. Adding a 10K pull down resistor onto EESK will cause the FT245BL to use USB Product ID 5005 (hex) instead of 5001 (hex). All of the other USB device descriptors are unchanged. *** Note 3</td>
</tr>
<tr>
<td>2</td>
<td>EEDATA</td>
<td>I/O</td>
<td>EEPROM – Data I/O Connected directly to Data-In of the EEPROM and to Data-Out of the EEPROM via a 2.2K resistor. Also pull Data-Out of the EEPROM to VCC via a 10K resistor for correct operation. *** Note 3</td>
</tr>
</tbody>
</table>
SX48BD/SX52BD
Configurable Communications Controllers with EE/Flash Program Memory, In-System Programming Capability, and On-Chip Debug

1.0 PRODUCT OVERVIEW

1.1 Introduction

The Ubicom SX48BD/SX52BD are members of the SX family of configurable communications controllers fabricated in an advanced CMOS process technology. The advanced process, combined with a RISC-based architecture, allows high-speed computation, flexible I/O control, and efficient data manipulation. Throughput is enhanced by operating the device at frequencies up to 75 MHz and by optimizing the instruction set to include mostly single-cycle instructions. In addition, the SX architecture is deterministic and totally reprogrammable. The Ubicom SX48BD and SX52BD are functionally the same, except for the package type and pinout. The SX48BD has four fewer pins and has only four rather than eight I/O pins for Port A.

On-chip functions include two 16-bit timers with 8-bit prescalers supporting different operating modes (PVM), simultaneous PWM capture, and external event counter, a general-purpose 8-bit timer with prescaler, an analog comparator, a brown-out detector, a wake-up timer, a power-save mode with multi-source wakeability, an internal RC oscillator, user-selectable clock modes, and high-current outputs.

The SX48BD and SX52BD are designed to replace traditional hardware functions.

Figure 1-1. Block Diagram

<table>
<thead>
<tr>
<th>Pin#</th>
<th>Signal</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>3V3OUT</td>
<td>OUT</td>
<td>3.3 volt Output from the integrated LDO regulator. This pin should be decoupled to GND using a 330uF ceramic capacitor in close proximity to the device pin. Its primary purpose is to provide external 3V3 supply to the USB transceiver cell and the RSTOUT pin. A small amount of current (&lt;= 5mA) can be drawn from this pin to power external 3.3V logic if required.</td>
</tr>
<tr>
<td>2,26</td>
<td>VCC</td>
<td>PWR</td>
<td>+4.35 volt or +5.25 volt VCC to the device core, LDO and none-FIFO interface pins.</td>
</tr>
<tr>
<td>13</td>
<td>VCCIO</td>
<td>PWR</td>
<td>+3.3 volt or +5.25 volt VCC to the FIFO interface pins 10, 12, 14, 16 and 18. 25. When interfacing with 3.3V external logic in a bus powered design connect VCCIO to a 3.3V supply generated from the USB bus. When interfacing with 3.3V external logic in a self powered design connect VCCIO to the 3.3V supply of the external logic. Otherwise connect to VCC to drive out at 5V CMOS level.</td>
</tr>
<tr>
<td>9,17</td>
<td>GND</td>
<td>PWR</td>
<td>Device - Ground Supply Pins</td>
</tr>
<tr>
<td>39</td>
<td>AVCC</td>
<td>PWR</td>
<td>Device - Analog Power Supply for the internal 8-bit clock multiplier</td>
</tr>
<tr>
<td>29</td>
<td>AGND</td>
<td>PWR</td>
<td>Device - Analog Ground Supply for the internal 8-bit clock multiplier</td>
</tr>
</tbody>
</table>

Note 1: In Input Mode, these pins are pulled to VCCIO via internal 20K ohm resistors. These can be programmed to gently pull down during USB suspend (PWREN = "1") by setting this option in the EEPROM.
Note 2: During device reset, these pins are Tri-state but pulled up to VCCIO via internal 20K ohm resistors.
Note 3: During device reset, these pins are Tri-state but pulled up to VCC via internal 20K ohm resistors.
2.0 CONNECTION DIAGRAMS

2.1 Pin Assignments

1.2 Key Features
75 MIPS Performance
- DC: 75 MHz operation
- 13.3 ns instruction cycle, 39.9 ns internal interrupt response at 75 MHz
- 1 instruction per clock (branches 3)

EE/FLASH Program Memory and SRAM Data Memory
- Access time of < 13.3 ns provides single cycle access
- EE/Flash rated for > 10,000 rewrite cycles
- 4096 Words of EE/Flash program memory
- 262x8 bits SRAM data memory

CPU Features
- Compact instruction set
- All instructions are single cycle except branch
- Eight-level push/pop hardware stack for subroutine linkage
- Fast table lookup capability through run-time readable code (IREAD instruction)
- Predictable program execution flow for hard real-time applications

Fast and Deterministic Interrupt
- Jitter-free 3-cycle internal interrupt response
- Hardware context save/restore of key resources such as PC, W, STATUS, and FSR within the 3-cycle interrupt response time
- External wake-up interrupt capability on Port B (8 pins)

Flexible I/O
- All pins individually programmable as I/O
- Inputs are TTL or CMOS level selectable
- All pins have selectable internal pull-ups
- Selectable Schmitt Trigger inputs on Ports B, C, D, and E
- All outputs capable of sourcing/shunting 30 mA
- Port A outputs have symmetrical drive
- Analog comparator support on Port B (RB0 OUT, RB1 IN, RB2 IN+)
- Selectable I/O operation synchronous to the oscillator clock

Hardware Peripheral Features
- Two 16-bit timers with 8-bit prescalers supporting:
  - Software Timer mode
  - PWM mode
  - Simultaneous PWM/Capture mode
  - External Event mode
- One 8-bit Real Time Clock/Counter (RTCC) with programmable 8-bit prescaler
- Watchdog Timer (shares the RTCC prescaler)
- Analog comparator
- Brown-out detector
- Multi-input Wakeup logic on 8 pins
- Internal RC oscillator with configurable rate (31.25 kHz to 4 MHz)
- Power-On-Reset

Packages
- 48-pin Tiny PQFP, and 52-pin PQFP

Programming and Debugging Support
- On-chip In-system programming support with serial or parallel interface
- In-system serial programming via oscillator pins
- On-chip In-system debugging support logic
- Real-time emulation, full program debug, and integrated development environment offered by third party tool vendors

Software Support
- Library of off-the-shelf Virtual Peripheral modules
- Examples of Virtual Peripheral integration
- Evaluation kits for communication intensive applications
### 2.2 Pin Descriptions

<table>
<thead>
<tr>
<th>Name</th>
<th>Pin Type</th>
<th>Input Levels</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RA0</td>
<td>I/O</td>
<td>TTL/CMOS</td>
<td>Bidirectional I/O Pin, symmetrical source / sink capability</td>
</tr>
<tr>
<td>RA1</td>
<td>I/O</td>
<td>TTL/CMOS</td>
<td>Bidirectional I/O Pin, symmetrical source / sink capability</td>
</tr>
<tr>
<td>RA2</td>
<td>I/O</td>
<td>TTL/CMOS</td>
<td>Bidirectional I/O Pin, symmetrical source / sink capability</td>
</tr>
<tr>
<td>RA3</td>
<td>I/O</td>
<td>TTL/CMOS</td>
<td>Bidirectional I/O Pin, symmetrical source / sink capability</td>
</tr>
<tr>
<td>RA4</td>
<td>I/O</td>
<td>TTL/CMOS</td>
<td>Bidirectional I/O Pin, symmetrical source / sink capability (52-pin pkg. only)</td>
</tr>
<tr>
<td>RA5</td>
<td>I/O</td>
<td>TTL/CMOS</td>
<td>Bidirectional I/O Pin, symmetrical source / sink capability (52-pin pkg. only)</td>
</tr>
<tr>
<td>RA6</td>
<td>I/O</td>
<td>TTL/CMOS</td>
<td>Bidirectional I/O Pin, symmetrical source / sink capability (52-pin pkg. only)</td>
</tr>
<tr>
<td>RA7</td>
<td>I/O</td>
<td>TTL/CMOS</td>
<td>Bidirectional I/O Pin, symmetrical source / sink capability (52-pin pkg. only)</td>
</tr>
<tr>
<td>RB0</td>
<td>I/O</td>
<td>TTL/CMOS/SST</td>
<td>Bidirectional I/O Pin, comparator output, MIWU/Interrupt input</td>
</tr>
<tr>
<td>RB1</td>
<td>I/O</td>
<td>TTL/CMOS/SST</td>
<td>Bidirectional I/O Pin, comparator negative input, MIWU/Interrupt input</td>
</tr>
<tr>
<td>RB2</td>
<td>I/O</td>
<td>TTL/CMOS/SST</td>
<td>Bidirectional I/O Pin, comparator positive input, MIWU/Interrupt input</td>
</tr>
<tr>
<td>RB3</td>
<td>I/O</td>
<td>TTL/CMOS/SST</td>
<td>Bidirectional I/O Pin, MIWU/Interrupt input</td>
</tr>
<tr>
<td>RB4</td>
<td>I/O</td>
<td>TTL/CMOS/SST</td>
<td>Bidirectional I/O Pin, MIWU/Interrupt input, Timer T1 Capture Input 1</td>
</tr>
<tr>
<td>RB5</td>
<td>I/O</td>
<td>TTL/CMOS/SST</td>
<td>Bidirectional I/O Pin, MIWU/Interrupt input, Timer T1 Capture Input 2</td>
</tr>
<tr>
<td>RB6</td>
<td>I/O</td>
<td>TTL/CMOS/SST</td>
<td>Bidirectional I/O Pin, MIWU/Interrupt input, Timer T1 PWM/Compare Output</td>
</tr>
<tr>
<td>RB7</td>
<td>I/O</td>
<td>TTL/CMOS/SST</td>
<td>Bidirectional I/O Pin, MIWU/Interrupt input, Timer T1 External Event Input</td>
</tr>
<tr>
<td>RC0</td>
<td>I/O</td>
<td>TTL/CMOS/SST</td>
<td>Bidirectional I/O pin, Timer T2 Capture Input 1</td>
</tr>
<tr>
<td>RC1</td>
<td>I/O</td>
<td>TTL/CMOS/SST</td>
<td>Bidirectional I/O pin, Timer T2 Capture Input 2</td>
</tr>
<tr>
<td>RC2</td>
<td>I/O</td>
<td>TTL/CMOS/SST</td>
<td>Bidirectional I/O pin, Timer T2 PWM/Compare Output</td>
</tr>
<tr>
<td>RC3</td>
<td>I/O</td>
<td>TTL/CMOS/SST</td>
<td>Bidirectional I/O pin, Timer T2 External Event Counter Input</td>
</tr>
<tr>
<td>RC4</td>
<td>I/O</td>
<td>TTL/CMOS/SST</td>
<td>Bidirectional I/O pin</td>
</tr>
<tr>
<td>RC5</td>
<td>I/O</td>
<td>TTL/CMOS/SST</td>
<td>Bidirectional I/O pin</td>
</tr>
<tr>
<td>RC6</td>
<td>I/O</td>
<td>TTL/CMOS/SST</td>
<td>Bidirectional I/O pin</td>
</tr>
<tr>
<td>RC7</td>
<td>I/O</td>
<td>TTL/CMOS/SST</td>
<td>Bidirectional I/O pin</td>
</tr>
<tr>
<td>RD0</td>
<td>I/O</td>
<td>TTL/CMOS/SST</td>
<td>Bidirectional I/O pin</td>
</tr>
<tr>
<td>RD1</td>
<td>I/O</td>
<td>TTL/CMOS/SST</td>
<td>Bidirectional I/O pin</td>
</tr>
<tr>
<td>RD2</td>
<td>I/O</td>
<td>TTL/CMOS/SST</td>
<td>Bidirectional I/O pin</td>
</tr>
<tr>
<td>RD3</td>
<td>I/O</td>
<td>TTL/CMOS/SST</td>
<td>Bidirectional I/O pin</td>
</tr>
<tr>
<td>RD4</td>
<td>I/O</td>
<td>TTL/CMOS/SST</td>
<td>Bidirectional I/O pin</td>
</tr>
<tr>
<td>RD5</td>
<td>I/O</td>
<td>TTL/CMOS/SST</td>
<td>Bidirectional I/O pin</td>
</tr>
<tr>
<td>RD6</td>
<td>I/O</td>
<td>TTL/CMOS/SST</td>
<td>Bidirectional I/O pin</td>
</tr>
<tr>
<td>RD7</td>
<td>I/O</td>
<td>TTL/CMOS/SST</td>
<td>Bidirectional I/O pin</td>
</tr>
<tr>
<td>RE0</td>
<td>I/O</td>
<td>TTL/CMOS/SST</td>
<td>Bidirectional I/O pin</td>
</tr>
<tr>
<td>RE1</td>
<td>I/O</td>
<td>TTL/CMOS/SST</td>
<td>Bidirectional I/O pin</td>
</tr>
<tr>
<td>RE2</td>
<td>I/O</td>
<td>TTL/CMOS/SST</td>
<td>Bidirectional I/O pin</td>
</tr>
<tr>
<td>RE3</td>
<td>I/O</td>
<td>TTL/CMOS/SST</td>
<td>Bidirectional I/O pin</td>
</tr>
<tr>
<td>RE4</td>
<td>I/O</td>
<td>TTL/CMOS/SST</td>
<td>Bidirectional I/O pin</td>
</tr>
<tr>
<td>RE5</td>
<td>I/O</td>
<td>TTL/CMOS/SST</td>
<td>Bidirectional I/O pin</td>
</tr>
<tr>
<td>RE6</td>
<td>I/O</td>
<td>TTL/CMOS/SST</td>
<td>Bidirectional I/O pin</td>
</tr>
<tr>
<td>RE7</td>
<td>I/O</td>
<td>TTL/CMOS/SST</td>
<td>Bidirectional I/O pin</td>
</tr>
<tr>
<td>RTCC</td>
<td>I</td>
<td>ST</td>
<td>Input to Real-Time Clock/Counter</td>
</tr>
<tr>
<td>MCLR</td>
<td>I</td>
<td>ST</td>
<td>Master Clear reset input – active low</td>
</tr>
<tr>
<td>OSC1/NAVp</td>
<td>I</td>
<td>ST</td>
<td>Crystal oscillator input – external clock source input</td>
</tr>
<tr>
<td>OSC2/Out</td>
<td>O</td>
<td>CMOS</td>
<td>Crystal oscillator output – in RC mode, internally pulled to VDD through weak pull-up</td>
</tr>
<tr>
<td>Vdd</td>
<td>P</td>
<td>–</td>
<td>Positive supply pins (a total of four positive supply pins, one on each side of the device)</td>
</tr>
<tr>
<td>Vss</td>
<td>P</td>
<td>–</td>
<td>Ground pins (a total of four ground pins, one on each side of the device)</td>
</tr>
</tbody>
</table>

Note: I = input, O = output, I/O = Input/Output, P = Power, TTL = TTL input, CMOS = CMOS input.
LAMPIRAN D

FOTO ALAT

D-1
Gambar D.1 Maket Tampak Samping

Gambar D.2 Maket Tampak Atas
Lampiran

Gambar D.3 Modul Tampak Atas

Gambar D.4 Modul Tampak Samping