MODULE 24: WEBCAM

On Completion of this module you will be able to use a WebCam with your programs, either to take a photo or monitoring purposes, such as security or motion detection.

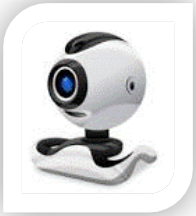
Subject Outcome 1: Introduction

Subject Outcome 2: Viewer

Subject Outcome 3: Take a photo of WebCam Preview

Subject Outcome 4: Motion Detection

Subject Outcome 5: QWebCam (recommended method, unless it clashes with your internal coding)



24.1 INTRODUCTION

With all modern requirements, some form of ID or photo is required by your program. It is rather frustrating having to open a 3rd party program, take a photo and then copy/paste or add to the original program. It is therefore ideal to have a program that has this function build in as required by the user. Please remember that the response speed of your camera is influenced by the quality and type of camera (HD high speed, VGA slow speed, etc.)

24.2 VIEWER

We will now code a WebCam and use it within the program. Settings such as size, pixel format, etc. will be added as we continue coding the functions accordingly.

SNO	CODING	EXPLANATION
1	$TYPECHECK ON $include "rapidq2.inc" declare sub doallactions CONST WS_VISIBLE = &H10000000 CONST WS_CHILD = &H40000000 CONST WM_USER = 1024 CONST WM_CAP_DRIVER_CONNECT = WM_USER + 10 CONST WM_CAP_DRIVER_DISCONNECT = WM_USER + 11 CONST WM_CAP_FILE_SAVEDIB = WM_USER + 25 CONST WM_CAP_SET_PREVIEW = WM_USER + 50 CONST WM_CAP_SET_PREVIEWRATE = WM_USER + 52 CONST WM_CAP_SET_SCALE = WM_USER + 53 const WM_CAPFIRST = 1024 Const WM_CAP_GRAB_FRAME = WM_CAPFIRST + 60 Const WM_CAP_DLG_VIDEOFORMAT = WM_CAPFIRST + 41 Const WM_CAP_DLG_VIDEOSOURCE = WM_CAPFIRST + 42 Const HWND_TOP = 0 Const SWP_NOACTIVATE = &H10 Const SWP_SHOWWINDOW = &H40 Declare Function CallAsmProc LIB "user32" Alias "CallWindowProcA" (Proc AS Long, A1 AS Long, A2 AS Long, A3 AS Long, A4 AS Long) AS Long Declare Function SendMessageX Lib "user32" Alias "SendMessageA" (hWnd As Long, wMsg As Long, wParam As Long, lParam As Long) As Long Declare Function capCreateCaptureWindow Lib "avicap32.dll" Alias "capCreateCaptureWindowA" (lpszWindowName As String, dwStyle As Long, left As Long, top As Long, nWidth As Long, nHeight As Long, hwndParent As Long, nID As Long) As Long Declare Function Beep Lib "kernel32" Alias "Beep" (ByVal dwFreq As Long, ByVal dwDuration As Long) As Long DECLARE SUB OnClose_frmMyForm DECLARE FUNCTION CreateViewWindow() AS LONG DEFINT iReturn DEFINT iDevice = 0 DEFINT iFramesSpacedMS = 10 DEFSTR sFileName , sTmp CREATE frmMyForm AS QFORM Center : Width = 1050 : height = 600 : borderstyle = 4 Caption = " POWERVIEW vs 1.1" AutoScroll = FALSE OnClose = OnClose_frmMyForm	Establish and display the WebCam preview Clear memory and close. Device nr. FPS (speed)

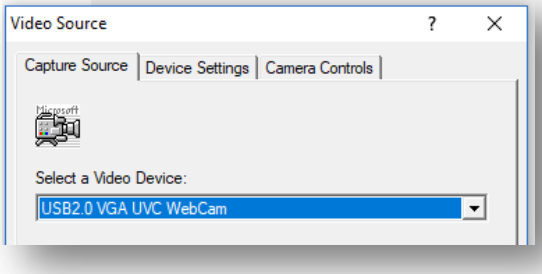
SNO	CODING	EXPLANATION

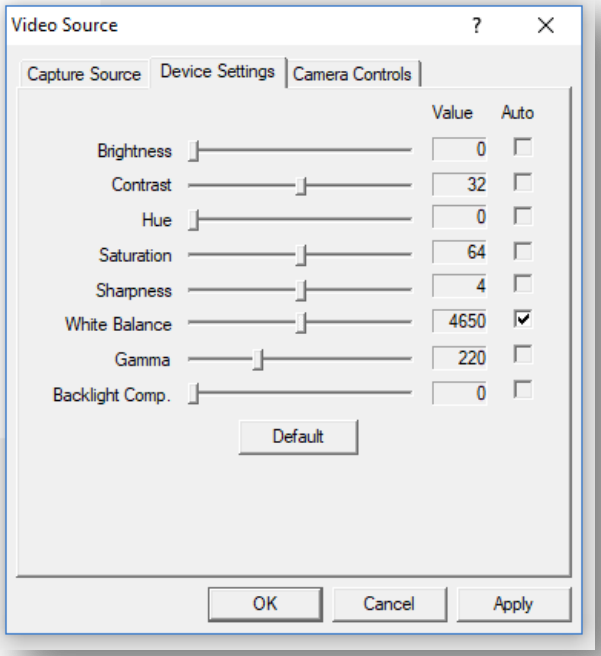
	CREATE rchWin AS QRICHEDIT
	Height = 200 : Align = alTop : visible = 0
	font.Name = "courier" : Font.Size = 8 : color = &HFAFFFF
	font.Color = &H000040 : ReadOnly = True : WordWrap = FALSE
	ScrollBars = ssBoth : HideSelection = False : PlainText = TRUE
	END CREATE
	END CREATE
	setwindowlong(frmMyForm.Handle , GWL_HWNDPARENT , HWND_DESKTOP)
	setwindowlong(application.Handle , GWL_HWNDPARENT , frmMyForm.Handle)
	DEFINT hCamWnd
	DEFSTR CamTitle = "CaptureWindow"
	DEFINT iScalable = FALSE
	frmMyForm.Show
	frmMyForm.Visible = FALSE
	doallactions
	frmMyForm.ShowModal
	
	SUB OnClose_frmMyForm
	DEFINT iReturn
	iReturn = SendMessageX(hCamWnd , WM_CAP_DRIVER_DISCONNECT , CamTitle , 0)
	end
	END SUB
	
	SUB doallactions
	DEFINT iTmp
	hCamWnd = capCreateCaptureWindow(CamTitle , WS_CHILD OR WS_VISIBLE , 0 , 0 , 4 * (frmMyForm.ClientHeight - 250) / 3 , frmMyForm.ClientHeight - 250 , frmMyForm.Handle , 0)
	IF hCamWnd <> 0 THEN
	rchWin.AddString(" - OK : Cam Opened")
	ELSE
	rchWin.AddString(" - ERROR : Camnot Opened Cam")
	END IF
	rchWin.AddString(" CONNECT = " & STR$(SendMessageX(hCamWnd , WM_CAP_DRIVER_CONNECT , iDevice , 0)))
	rchWin.AddString("SET REVIEW RATE = " & STR$(SendMessageX(hCamWnd , WM_CAP_SET_PREVIEWRATE , iFramesSpacedMS , 0)))
	rchWin.AddString("SET PREVIEW ON = " & STR$(SendMessageX(hCamWnd , WM_CAP_SET_PREVIEW , TRUE , 0)))
	END SUB
		This is required to communicate with the WebCam. Use as is.
		
		Activate the WebCam.
		
		Call all required actions.
		
		Clear the memory and exit.
		
		Position of the WebCam Preview on the Form.
		
		Communication with WebCam to be displayed.

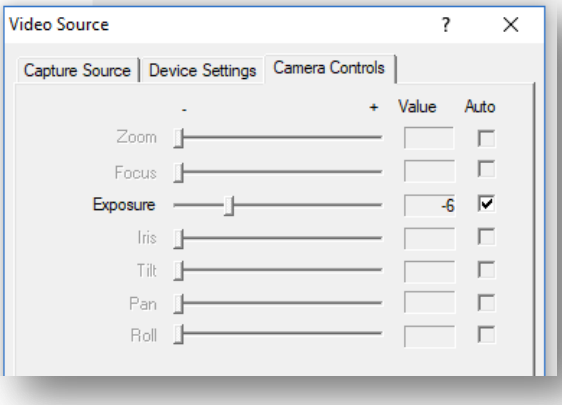
2	$TYPECHECK ON $include "rapidq2.inc" declare sub doallactions declare sub sizecamnow2 declare sub sizecamnow CONST WS_VISIBLE = &H10000000 CONST WS_CHILD = &H40000000 CONST WM_USER = 1024 CONST WM_CAP_DRIVER_CONNECT = WM_USER + 10 CONST WM_CAP_DRIVER_DISCONNECT = WM_USER + 11 CONST WM_CAP_FILE_SAVEDIB = WM_USER + 25 CONST WM_CAP_SET_PREVIEW = WM_USER + 50 CONST WM_CAP_SET_PREVIEWRATE = WM_USER + 52 CONST WM_CAP_SET_SCALE = WM_USER + 53 const WM_CAPFIRST = 1024 Const WM_CAP_GRAB_FRAME = WM_CAPFIRST + 60 Const WM_CAP_DLG_VIDEOFORMAT = WM_CAPFIRST + 41 Const WM_CAP_DLG_VIDEOSOURCE = WM_CAPFIRST + 42 Const HWND_TOP = 0 Const SWP_NOACTIVATE = &H10 Const SWP_SHOWWINDOW = &H40 Declare Function CallAsmProc LIB "user32" Alias "CallWindowProcA" (Proc AS Long, A1 AS Long, A2 AS Long, A3 AS Long, A4 AS Long) AS Long Declare Function SendMessageX Lib "user32" Alias "SendMessageA" (hWnd As Long, wMsg As Long, wParam As Long, lParam As Long) As Long Declare Function capCreateCaptureWindow Lib "avicap32.dll" Alias "capCreateCaptureWindowA" (lpszWindowName As String, dwStyle As Long, left As Long, top As Long, nWidth As Long, nHeight As Long, hwndParent As Long, nID As Long) As Long Declare Function Beep Lib "kernel32" Alias "Beep" (ByVal dwFreq As Long, ByVal dwDuration As Long) As Long	Set Size and Properties.

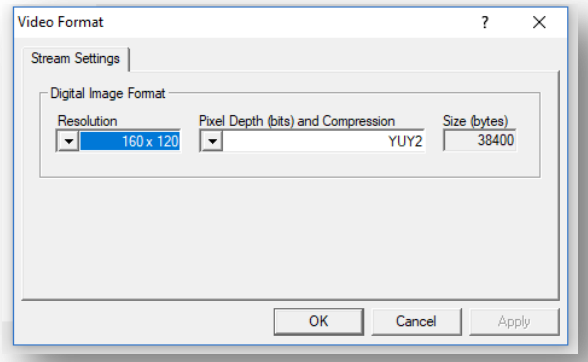
SNO	CODING	EXPLANATION

	DECLARE SUB OnClose_frmMyForm
	DECLARE FUNCTION CreateViewWindow() AS LONG
	DEFINT iReturn
	DEFINT iDevice = 0
	DEFINT iFramesSpacedMS = 10
	DEFSTR sFileName , sTmp
	CREATE frmMyForm AS QFORM
	Center : Width = 1050 : height = 600 : borderstyle = 4
	Caption = " POWERVIEW vs 1.1"
	AutoScroll = FALSE
	OnClose = OnClose_frmMyForm
	create menu as qmainmenu
	create menu1 as qmenuitem
	caption="size":onclick=sizecamnow
	end create
	create menu2 as qmenuitem
	caption="settings":onclick=sizecamnow2
	end create
	end create
	CREATE rchWin AS QRICHEDIT
	Height = 200 : Align = alTop : visible = 0
	font.Name = "courier" : Font.Size = 8 : color = &HFAFFFF
	font.Color = &H000040 : ReadOnly = True : WordWrap = FALSE
	ScrollBars = ssBoth : HideSelection = False : PlainText = TRUE
	END CREATE
	END CREATE
	setwindowlong(frmMyForm.Handle , GWL_HWNDPARENT , HWND_DESKTOP)
	setwindowlong(application.Handle , GWL_HWNDPARENT , frmMyForm.Handle)
	DEFINT hCamWnd
	DEFSTR CamTitle = "CaptureWindow"
	DEFINT iScalable = FALSE
	frmMyForm.Show
	frmMyForm.Visible = FALSE
	doallactions
	frmMyForm.ShowModal
	
	SUB OnClose_frmMyForm
	DEFINT iReturn
	iReturn = SendMessageX(hCamWnd , WM_CAP_DRIVER_DISCONNECT , CamTitle , 0)
	end
	END SUB
	
	SUB doallactions
	DEFINT iTmp
	hCamWnd = capCreateCaptureWindow(CamTitle , WS_CHILD OR WS_VISIBLE , 0 , 0 , 4 * (frmMyForm.ClientHeight - 250) / 3 , frmMyForm.ClientHeight - 250 , frmMyForm.Handle , 0)
	IF hCamWnd <> 0 THEN
	rchWin.AddString(" - OK : Cam Opened")
	ELSE
	rchWin.AddString(" - ERROR : Camnot Opened Cam")
	END IF
	rchWin.AddString(" CONNECT = " & STR$(SendMessageX(hCamWnd , WM_CAP_DRIVER_CONNECT , iDevice , 0)))
	rchWin.AddString("SET REVIEW RATE = " & STR$(SendMessageX(hCamWnd , WM_CAP_SET_PREVIEWRATE , iFramesSpacedMS , 0)))
	rchWin.AddString("SET PREVIEW ON = " & STR$(SendMessageX(hCamWnd , WM_CAP_SET_PREVIEW , TRUE , 0)))
	END SUB
	
	sub sizecamnow
	rchWin.Addstring ("VIDEO FORMAT SOURCE = " & Str$(SendMessageX (hCamWnd, WM_CAP_DLG_VIDEOSOURCE, 0, 0)) & " (If 0, You Have None)")
	end sub
	
	sub sizecamnow2
	rchWin.Addstring ("VIDEO FORMAT DIALOG = " & Str$(SendMessageX (hCamWnd, WM_CAP_DLG_VIDEOFORMAT, 0, 0)) & " (If 0, You Have None)")
	end sub
		Menu buttons to call size and property settings of the camera (if available).
		
		Size settings.
		
		Property settings.


Video Source
Capture Source
Device Settings
Camera Controls
Select a Video Device:
USB2.0 VGA UVC WebCam


Video Source
Capture Source
Device Settings
Camera Controls
Value
Auto
Brightness
0
Contrast
32
Hue
0
Saturation
64
Sharpness
4
White Balance
4650
Gamma
220
Backlight Comp.
0
Default
OK
Cancel
Apply


Video Source
Capture Source
Device Settings
Camera Controls
-
+
Value
Auto
Zoom
Focus
Exposure
-6
Iris
Tilt
Pan
Roll



Access to the abovementioned settings differ from camera to camera as some may zoom, etc. If possible, just buy a HD camera for best possible quality and speed (no delay on preview).

24.3 TAKE PHOTO WITH WEB-CAM

A photo will be taken and recorded as a BMP file. You may use QIMAGEEX and QBITMAPEX to convert as a JPG file if you require it to be in that format. You will save the file directly from the Web-Cam communicator (should the commands not execute, replace the QRICHEDIT with QLISTBOX and instead of ADDSTRINGS, use ADDITEMS. It is much more effective as the conversion of the coding is clean without WORDPAD characters influences.

SNO	CODING	EXPLANATION
1	$TYPECHECK ON $include "rapidq2.inc" declare sub doallactions declare sub sizecamnow2 declare sub sizecamnow declare sub takephoto CONST WS_VISIBLE = &H10000000 CONST WS_CHILD = &H40000000 CONST WM_USER = 1024 CONST WM_CAP_DRIVER_CONNECT = WM_USER + 10 CONST WM_CAP_DRIVER_DISCONNECT = WM_USER + 11 CONST WM_CAP_FILE_SAVEDIB = WM_USER + 25 CONST WM_CAP_SET_PREVIEW = WM_USER + 50 CONST WM_CAP_SET_PREVIEWRATE = WM_USER + 52 CONST WM_CAP_SET_SCALE = WM_USER + 53 const WM_CAPFIRST = 1024 Const WM_CAP_GRAB_FRAME = WM_CAPFIRST + 60 Const WM_CAP_DLG_VIDEOFORMAT = WM_CAPFIRST + 41 Const WM_CAP_DLG_VIDEOSOURCE = WM_CAPFIRST + 42 Const HWND_TOP = 0 Const SWP_NOACTIVATE = &H10 Const SWP_SHOWWINDOW = &H40 Declare Function CallAsmProc LIB "user32" Alias "CallWindowProcA" (Proc AS Long, A1 AS Long, A2 AS Long, A3 AS Long, A4 AS Long) AS Long Declare Function SendMessageX Lib "user32" Alias "SendMessageA" (hWnd As Long, wMsg As Long, wParam As Long, lParam As Long) As Long	

SNO	CODING	EXPLANATION

	Declare Function capCreateCaptureWindow Lib "avicap32.dll" Alias "capCreateCaptureWindowA" (lpszWindowName As String, dwStyle As Long, left As Long, top As Long, nWidth As Long, nHeight As Long, hwndParent As Long, nID As Long) As Long
	Declare Function Beep Lib "kernel32" Alias "Beep" (ByVal dwFreq As Long, ByVal dwDuration As Long) As Long
	DECLARE SUB OnClose_frmMyForm
	DECLARE FUNCTION CreateViewWindow() AS LONG
	dim milda as string
	DEFINT iReturn
	DEFINT iDevice = 0
	DEFINT iFramesSpacedMS = 10
	DEFSTR sFileName , sTmp
	CREATE frmMyForm AS QFORM
	Center : Width = 1050 : height = 600 : borderstyle = 4
	Caption = " POWERVIEW vs 1.1"
	AutoScroll = FALSE
	OnClose = OnClose_frmMyForm
	create menu as qmainmenu
	create menu1 as qmenuitem
	caption="size":onclick=sizecamnow
	end create
	create menu2 as qmenuitem
	caption="settings":onclick=sizecamnow2
	end create
	create menu3 as qmenuitem
	caption="TAKE PHOTO":onclick=takephoto
	end create
	end create
	create photo as qimage
	left=700:top=10:autosize=1
	end create
	CREATE rchWin AS QRICHEDIT
	Height = 200 : Align = alTop : visible = 0
	font.Name = "courier" : Font.Size = 8 : color = &HFAFFFF
	font.Color = &H000040 : ReadOnly = True : WordWrap = FALSE
	ScrollBars = ssBoth : HideSelection = False : PlainText = TRUE
	END CREATE
	END CREATE
	setwindowlong(frmMyForm.Handle , GWL_HWNDPARENT , HWND_DESKTOP)
	setwindowlong(application.Handle , GWL_HWNDPARENT , frmMyForm.Handle)
	DEFINT hCamWnd
	DEFSTR CamTitle = "CaptureWindow"
	DEFINT iScalable = FALSE
	frmMyForm.Show
	frmMyForm.Visible = FALSE
	doallactions
	frmMyForm.ShowModal
	
	SUB OnClose_frmMyForm
	DEFINT iReturn
	iReturn = SendMessageX(hCamWnd , WM_CAP_DRIVER_DISCONNECT , CamTitle , 0)
	end
	END SUB
	
	SUB doallactions
	DEFINT iTmp
	hCamWnd = capCreateCaptureWindow(CamTitle , WS_CHILD OR WS_VISIBLE , 0 , 0 , 4 * (frmMyForm.ClientHeight - 250) / 3 , frmMyForm.ClientHeight - 250 , frmMyForm.Handle , 0)
	IF hCamWnd <> 0 THEN
	rchWin.AddString(" - OK : Cam Opened")
	ELSE
	rchWin.AddString(" - ERROR : Camnot Opened Cam")
	END IF
	rchWin.AddString(" CONNECT = " & STR$(SendMessageX(hCamWnd , WM_CAP_DRIVER_CONNECT , iDevice , 0)))
	rchWin.AddString("SET REVIEW RATE = " & STR$(SendMessageX(hCamWnd , WM_CAP_SET_PREVIEWRATE , iFramesSpacedMS , 0)))
	rchWin.AddString("SET PREVIEW ON = " & STR$(SendMessageX(hCamWnd , WM_CAP_SET_PREVIEW , TRUE , 0)))	

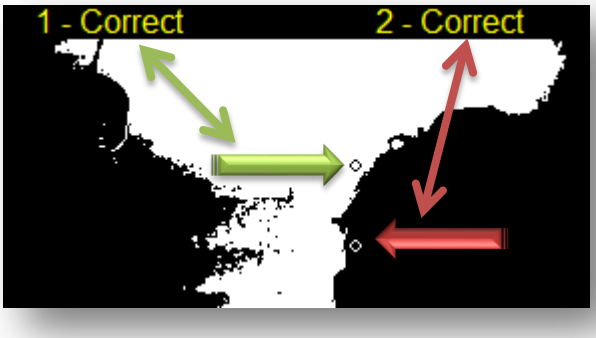
SNO	CODING	EXPLANATION
	END SUB sub sizecamnow rchWin.Addstring ("VIDEO FORMAT SOURCE = " & Str$(SendMessageX (hCamWnd, WM_CAP_DLG_VIDEOSOURCE, 0, 0)) & " (If 0, You Have None)") end sub sub sizecamnow2 rchWin.Addstring ("VIDEO FORMAT DIALOG = " & Str$(SendMessageX (hCamWnd, WM_CAP_DLG_VIDEOFORMAT, 0, 0)) & " (If 0, You Have None)") end sub sub takephoto milda="c:\ms3\~common\camtst.bmp" if fileexists(milda)=1 then kill milda sFileName = milda rchwin.addstrings(" FILE SAVE BMP = " & STR$(SendMessageX(hCamWnd , WM_CAP_FILE_SAVEDIB , 0 , VARPTR(sFileName))) & ""+chr$(13)+" In CurDir : " & "c:\ms3\~common\camtst.bmp") photo.bmp=milda end sub	Name the file, if exist, delete to create new file. Save to location. Display QIMAGE the photo that was taken.

24.4 MOTION DETECTION

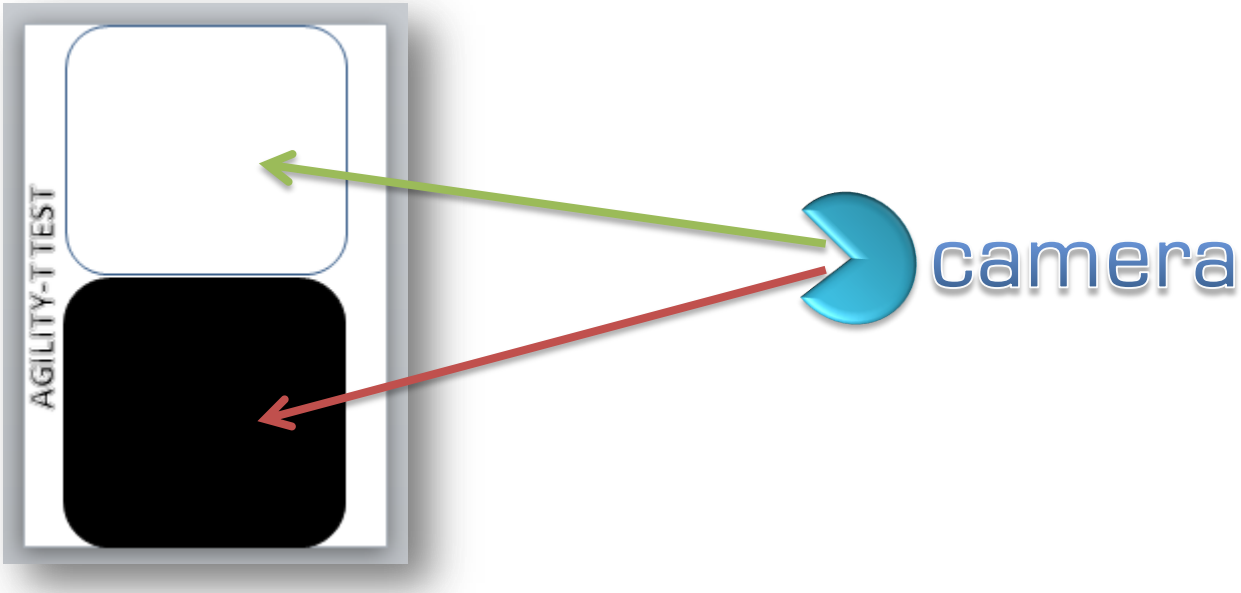
Motion detection is used for many purposes; security applications, start of execution, gesture control, etc. There are two main methods of monitoring motion, full screen alteration (pixel differences) or spot-difference. Spot-difference is to monitor specific pixels on the screen for any alteration in colour and full screen alteration is a complete shift in RGB colouring. We will address both methods during this lesson.

24.4.1 Spot-Difference

We will monitor only the specified pixels on the previewer using a monochrome screen (two colour screen). Notice the two focus rings. As soon as motion is detected the value (white top and black bottom) will alter therefore registering motion. For this to work you need to ensure that the background has a dark spot and light spot. This is ideal for motion motion detection related to a line (finish line – athletics, etc.)



Camera focus point would be for instance a card with the following printed:



SNO	CODING	EXPLANATION
1	$TYPECHECK ON $INCLUDE <RapidQ2.inc> dim rect2 as qrect dim CapBitMap1 as qbitmapex capbitmap1.pixelformat=1 capbitmap1.Monochrome=1 CONST WS_VISIBLE = &H10000000 CONST WS_CHILD = &H40000000 CONST WM_USER = 1024 CONST WM_CAP_DRIVER_CONNECT = WM_USER + 10 CONST WM_CAP_DRIVER_DISCONNECT = WM_USER + 11 CONST WM_CAP_FILE_SAVEDIB = WM_USER + 25 CONST WM_CAP_SET_PREVIEW = WM_USER + 50 CONST WM_CAP_SET_PREVIEWRATE = WM_USER + 52 CONST WM_CAP_SET_SCALE = WM_USER + 53 const WM_CAPFIRST = 1024 Const WM_CAP_GRAB_FRAME= WM_CAPFIRST + 60 Const WM_CAP_DLG_VIDEOFORMAT= WM_CAPFIRST + 41 Const WM_CAP_DLG_VIDEOSOURCE= WM_CAPFIRST + 42 Const HWND_TOP= 0 Const SWP_NOACTIVATE= &H10 Const SWP_SHOWWINDOW= &H40 DECLARE FUNCTION CreateViewWindow() AS LONG DECLARE FUNCTION CallAsmProc LIB "user32" ALIAS "CallWindowProcA"(Proc AS LONG , A1 AS LONG , A2 AS LONG , A3 AS LONG , A4 AS LONG) AS LONG DECLARE FUNCTION SendMessageX LIB "user32" ALIAS "SendMessageA"(hWnd AS LONG , wMsg AS LONG , wParam AS LONG , lParam AS LONG) AS LONG DECLARE FUNCTION capCreateCaptureWindow LIB "avicap32.dll" ALIAS "capCreateCaptureWindowA"(lpszWindowName AS STRING , dwStyle AS LONG , Left AS LONG , Top AS LONG , nWidth AS LONG , nHeight AS LONG , hwndParent AS LONG , nID AS LONG) AS LONG DECLARE FUNCTION Beep LIB "kernel32" ALIAS "Beep"(ByVal dwFreq AS LONG , ByVal dwDuration AS LONG) AS LONG DEFINT iReturn DEFINT iDevice = 0 DEFINT iFramesSpacedMS = 5 DEFSTR sFileName , sTmp	Required to copy the image. This will transform the image to monocolour for easy motion detection (black and white image).

SNO	CODING	EXPLANATION

	DIM font2k AS QFONT : font2k.Name = "arial" : font2k.Size = 12 : font2k.Color = 65535
	declare sub calibrateagil
	declare sub updatecalibrateagiltest
	declare sub doallactions
	dim calibrateagiltest as qtimer: calibrateagiltest.enabled=0:calibrateagiltest.interval=100
	calibrateagiltest.ontimer=updatecalibrateagiltest
	
	CREATE frmMyForm AS qformex
	FormStyle=fsStayOnTop:onshow=calibrateagil
	CREATE rchWin AS QRICHEDIT
	Height = 200 : Align = alTop : Visible = 0
	font.Name = "courier" : font.Size = 8 : Color = &HFAFFFF
	font.Color = &H000040 : ReadOnly = 1 : Wordwrap = FALSE
	ScrollBars = ssBoth : HideSelection = FALSE : PlainText = 1
	END CREATE
	AutoScroll = 1:BorderStyle = 1:Width = 1350:Height=725
	Top=10:Left=10:Enabled = 1:Color = 0:Visible = 0:borderstyle=4
	create agilcal as qimage
	left=680:top=20:autosize=1:visible=0
	end create
	create agilx1 as qlabel
	left=680:top=2:caption="1 - Correct":transparent=1:font=font2k
	end create
	create agilx2 as qlabel
	left=850:top=2:caption="2- Correct":transparent=1:font=font2k
	end create
	end create
	SetWindowLong(frmMyForm.Handle, -8, 0)
	SetWindowLong(Application.Handle, -8, frmMyForm.Handle)
	DEFINT hCamWnd
	DEFSTR CamTitle = "CaptureWindow"
	DEFINT iScalable = FALSE
	frmMyForm.Show
	frmMyForm.Visible = FALSE
	doallactions
	frmMyForm.ShowModal
	
	sub doallactions
	DEFINT iTmp
	hCamWnd = capCreateCaptureWindow(CamTitle , WS_CHILD OR WS_VISIBLE , 0, 2 , 650, 480,
	frmMyForm.Handle , 0)
	IF hCamWnd <> 0 THEN
	rchWin.AddString(" - OK : Cam Opened")
	ELSE
	rchWin.AddString(" - ERROR : Cannot Opened Cam")
	END IF
	rchWin.AddString(" CONNECT = " & STR$(SendMessageX(hCamWnd , WM_CAP_DRIVER_CONNECT ,
	iDevice , 0)))
	rchWin.AddString("SET REVIEW RATE = " & STR$(SendMessageX(hCamWnd ,
	WM_CAP_SET_PREVIEWRATE , iFramesSpacedMS , 0)))
	rchWin.AddString("SET PREVIEW ON = " & STR$(SendMessageX(hCamWnd ,
	WM_CAP_SET_PREVIEW , 1 , 0)))
	end sub
	
	sub calibrateagil
	frmMyForm.caption=" ACTIVE CAMERA ..."
	frmMyForm.width=1100:frmMyForm.height=500
	frmmyform.left=20:frmmyform.top=20
	agilcal.visible=1
	frmMyForm.visible=1
	rect2.left=frmmyform.left+8
	rect2.top=frmmyform.top+30
	rect2.right=rect2.left+320
	rect2.bottom=rect2.top+240
	screen.CaptureToBMP (rect2, CapBitMap1)
	agilcal.bmp=capbitmap1.bmp
	calibrateagiltest.enabled=1
	end sub
		Will calibrate the camera. Will check for motion. Motion timer check.
		
		The image will be redisplayed in mono-colour. Feedback spot 1.
		
		Feedback spot 2.
		
		Call the camera.
		
		Note the position.
		
		Calibrate the camera and set in action to wait for movement.
		
		Start the monitor of motion.

SNO	CODING	EXPLANATION

	sub updatecalibrateagiltest
	rect2.left=frmmyform.left+8
	rect2.top=frmmyform.top+30
	rect2.right=rect2.left+320
	rect2.bottom=rect2.top+240
	screen.CaptureToBMP (rect2, CapBitMap1)
	agilcal.bmp=capbitmap1.bmp
	agilcal.circle(157,60,163,66,0,-1)
	agilcal.circle(157,100,163,106,16777215,-1)
	if agilcal.pixel(160,63)=0 then
	agilx1.caption="1 - incorrect"
	end if
	if agilcal.pixel(160,103)=16777215 then
	agilx2.caption="2 - incorrect"
	end if
	if agilcal.pixel(160,63)=16777215 then
	agilx1.caption="1 - Correct"
	end if
	if agilcal.pixel(160,103)=0 then
	agilx2.caption="2 - Correct"
	end if
	okskipper912x:
	end sub
		Update what is displayed in the spot areas. Capture and transform as mono colour.
		
		Check if not white = execute if movement is picked up. Check if not black.
		
		Check if spot 1 is correct.
		
		Check if spot 2 is correct.

24.4.2 Full Screen monitor

With this method the entire image projected by the camera is monitored for extreme RGB colour changes. You may set the sensitivity accordingly. Use this coding as is, simply only execute your coding once the motion was detected (BEEP).

SNO	CODING

1	$ESCAPECHARS ON
	$TYPECHECK ON
	$INCLUDE "RapidQ2.Inc"
	Const SND_XP = 5
	Declare Function SND_GetVersion Lib "kernel32" Alias "GetVersion" () As Long
	Declare Function SND_Beep Lib "kernel32" Alias "Beep" (ByVal dwFreq As Long, ByVal dwDuration As Long) As Long
	DefInt SND_OS_Version = SND_GetVersion And 255
	Sub Sound___Ex (Freq As Long, Dura As Long)
	If SND_OS_Version < SND_XP Then
	Else
	End If
	End Sub
	$UNDEF Sound
	$DEFINE Sound Sound___Ex
	Const WS_VISIBLE= &H10000000:Const WS_CHILD= &H40000000
	Const WM_CAPFIRST = 1024
	Const WM_CAP_DRIVER_CONNECT= WM_CAPFIRST + 10
	Const WM_CAP_DRIVER_DISCONNECT= WM_CAPFIRST + 11
	Const WM_CAP_FILE_SAVEDIB= WM_CAPFIRST + 25
	Const WM_CAP_SET_SCALE= WM_CAPFIRST + 53
	Const WM_CAP_GRAB_FRAME= WM_CAPFIRST + 60
	Const WM_CAP_DLG_VIDEOFORMAT= WM_CAPFIRST + 41
	Const WM_CAP_DLG_VIDEOSOURCE= WM_CAPFIRST + 42
	Const HWND_TOP= 0
	Const SWP_NOACTIVATE= &H10
	Const SWP_SHOWWINDOW= &H40
	Declare Function CallAsmProc LIB "user32" Alias "CallWindowProcA" (Proc AS Long, A1 AS Long, _
	A2 AS Long, A3 AS Long, A4 AS Long) AS Long
	Declare Function SendMessageX Lib "user32" Alias "SendMessageA" (hWnd As Long, wMsg As Long, _
	wParam As Long, lParam As Long) As Long
	Declare Function capCreateCaptureWindow Lib "avicap32.dll" Alias "capCreateCaptureWindowA" _
	(lpszWindowName As String, dwStyle As Long, left As Long, top As Long, nWidth As Long, _
	nHeight As Long, hwndParent As Long, nID As Long) As Long

SNO	CODING

```
Declare Function Beep Lib "kernel32" Alias "Beep" (ByVal dwFreq As Long, ByVal dwDuration As Long) As Long
Declare Sub OnClose_frmDetectMotion
Declare Sub OnClic_btnStart
Declare Function CreateViewWindow () As Long
Declare Sub OnTimer_tmrSpy
Declare Sub OnKeypress_frmDetectMotion (key As Byte)
Declare Function GetOneFrame () As String
Declare Sub OnClic_AnyMenu (Sender As QMenuItem)
Create frmDetectMotion As QForm
  Center:Width = 680:height = 700:Color = &HC13F00
  Caption = "frmDetectMotion"
  KeyPreview = True:BorderStyle = bsSingle
  AutoScroll = False:OnKeyPress=OnKeypress_frmDetectMotion
  OnClose = OnClose_frmDetectMotion
  Create pnlBottom As QPanel
     Height = 35
     Align = alBottom
     Font.Name = "courier"
     Font.Size = 16
     Create lblThreshold As QLabel
        Align = alLeft
        Color = &HD7F2F0
        AutoSize = True
        ShowHint = True
        Hint = "Adjust Threshold with Key '+' and '-'\nto be just above Dpc maximum"
        Caption = " THRESHOLD: 2.5 pc (+/- change) "
     End Create
     Create btnStart AS QButton
       Align = alClient
       Font.Color = &H1E1EFF
       Font.Bold = True
       Caption = "&START"
       OnClick = OnClic_btnStart
     End Create
  End Create
  Create rchWin AS qlistbox
     Height = 150:Align = alTop
     Font.Name = "courier":Font.Size = 12
     color = &HFAFFFF:Font.Color = &H000040
     Text =  "\n JUST PLUG A CAM IN and CLICK ON \"START\"" _
       & "\n\n Then adjust Threshold with Key '+' and '-'"
  End Create
  Create mnuMain AS QMAINMENU
      Create mnuVideoSetup AS QMENUITEM
         Caption = "&CONFIG CAM"
         Create mnuFormatDlg AS QMENUITEM
            Caption = "&VIDEO FORMAT (Size)"
            OnClick = OnClic_AnyMenu
         End Create
         Create mnuSourceDlg AS QMENUITEM
            Caption = "&VIDEO SOURCE (Brightness, Contrast, ...)"
            OnClick = OnClic_AnyMenu
         End Create
      End Create
   End Create
End Create
' -- ARRAY containing ASM GetBMPDifference --
DefByte GetBMPDifferenceArray (0 To 88) = _
  {&H55,&H89,&HE5,&H8B,&H75,&H08,&H8B,&H7D,&H0C,&H8B,&H46,&H0A,&H8B,&H4F,&H0A,&H39, _
   &HC8,&H74,&H0A,&HB8,&H01,&H00,&H00,&H00,&HE9,&H36,&H00,&H00,&H00,&HFC,&H41,&H49, _
   &H74,&H0D,&HA6,&H74,&HFA,&HB8,&H02,&H00,&H00,&H00,&HE9,&H24,&H00,&H00,&H00,&H31, _
   &HC0,&H8B,&H75,&H08,&H8B,&H4E,&H22,&H03,&H76,&H0A,&H8B,&H7D,&H0C,&H03,&H7F,&H0A, _
   &H31,&HDB,&H8A,&H1E,&H8A,&H17,&H28,&HD3,&H73,&H02,&HF6,&HDB,&H01,&HD8,&H46,&H47, _
   &H49,&H75,&HEF,&H89,&HEC,&H5D,&HC2,&H10,&H00}
' -- POINTER to use In CallAsmProc --
DefInt ptrGetBMPDifference = VarPtr (GetBMPDifferenceArray(0))
' -- RQ CALL GetBMPDifference --
Function GetBMPDifference (ptrPic_1 As Long, ptrPic_2 As Long) As Long
   Result = CallAsmProc (ptrGetBMPDifference, ptrPic_1, ptrPic_2, 0, 0)
```

SNO	CODING

```
End Function

' GLOBAL DATAS
DefInt hCamWnd
DefStr CamTitle = "CaptureWindow"
DefInt iDevice = 0
Dim tmrSpy As QTimer
tmrSpy.Enabled = False
tmrSpy.Interval = 100
tmrSpy.OnTimer = OnTimer_tmrSpy
DefInt flagPic1or2, flagStarted = False
DefStr Bmp1 = "Pic1.Bmp", Bmp2 = "Pic2.Bmp"
DefStr strBmp1, strBmp2
DefInt iBmpDatasLength
DefInt iFrameCount = 0
DefInt iLastDiff = 0
Defint iThreshold = 400
DefInt flagMenuBusy = False
frmDetectMotion.ShowModal

Sub OnClose_frmDetectMotion
  SendMessageX (hCamWnd, WM_CAP_DRIVER_DISCONNECT, Camtitle, 0)
  Application.Terminate
End Sub

Function GetOneFrame () As String
   DefInt iGrabed = SendMessageX (hCamWnd, WM_CAP_GRAB_FRAME, 0, 0), iSaved
   Dim fileBmp As QFileStream
   DefStr sBmpFileName
   If iFrameCount Mod 2 = 1 Then
      sBmpFileName = Bmp1
   Else
      sBmpFileName = Bmp2
   End If
   iSaved = SendMessageX (hCamWnd, WM_CAP_FILE_SAVEDIB, 0, VarPtr(sBmpFileName))
   fileBmp.Open (sBmpFileName, fmOpenRead)
      If iFrameCount Mod 2 = 1 Then
         strBmp1 = fileBmp.ReadBinStr (fileBmp.Size)
      Else
         strBmp2 = fileBmp.ReadBinStr (fileBmp.Size)
      End If
   fileBmp.Close
   Result = "Frame=" & Str$(iFrameCount)
   If (iGrabed = 0 Or iSaved = 0) Then Result = Result & " Grab=" & Str$(iGrabed) & " Save=" &  Str$(iSaved)
   Inc iFrameCount
End Function

Sub OnClic_btnStart
   DefStr sTmp
   If flagStarted = True Then Exit Sub
   With frmDetectMotion
      hCamWnd = capCreateCaptureWindow(CamTitle,  ws_child + ws_visible, 0, 210, _
            4*(.ClientHeight-250)/3, .ClientHeight-250, .Handle, 0)
      SetWindowPos (hCamWnd, HWND_TOP, (.ClientWidth - 640)/2 , rchWin.Top + rchWin.Height  + _
            (.ClientHeight - 480 - rchWin.Top - rchWin.height - pnlBottom.Height)/2, 640, _
             480, SWP_NOACTIVATE Or SWP_SHOWWINDOW)
   End With
   If hCamWnd <> 0 then
      rchWin.additems ("\n - OK : Cam Opened")
      flagStarted = True
   Else
      rchWin.additems ("\n - ERROR : Cannot Opened Cam")
      flagStarted = 0
      Exit Sub
   End If
   If hCamWnd = 0 Then ShowMessage ("You must Open Cam First"):Exit Sub
   rchWin.additems ("        CONNECT = " & Str$(SendMessageX (hCamWnd, WM_CAP_DRIVER_CONNECT, iDevice, 0)) _
                                          & "  Device: " & Str$(iDevice))
```

SNO	CODING

```
    rchWin.additems ("    SET SCALABLE = " & Str$(SendMessageX (hCamWnd, WM_CAP_SET_SCALE, false, 0)))
    rchWin.additems GetOneFrame
    sTmp = GetOneFrame
    iLastDiff = GetBmpDifference (VarPtr(strBmp1), VarPtr(strBmp2))
    rchWin.additems ("Diff=" & Str$(iLastDiff) & " " & sTmp)
    tmrSpy.Enabled = True
End Sub

Sub OnTimer_tmrSpy
    DefStr sTmp = GetOneFrame, sBeep = ""
    MemCpy(VarPtr(iBmpDatasLength), VarPtr(strBmp1) + 34, 4)
    frmDetectMotion.Caption = "MOTION DETECTION - Image Size: " & Str$(iBmpDatasLength) & " bytes" _
                                        & " - GrabTime: " & Str$(tmrSpy.Interval) & " ms"
    DefSng sngStartChrono = Timer
    DefInt iDiff = GetBmpDifference (VarPtr(strBmp1), VarPtr(strBmp2))
    DefInt iDuration = Abs(1000 * (Timer - sngStartChrono))
    DefSng sngThreshold = iThreshold/1000
    If (iDiff > iLastDiff * (1 + sngThreshold) Or iDiff < iLastDiff * (1 - sngThreshold)) Then
       Sound (1000, 8):Beep(1000, 300)
       sBeep = " BEEP"
    End If
    rchWin.additems ("Diff=" & Str$(iDiff) _
            & " Dpc=" & Strf$(Abs(100*(iLastDiff-iDiff)/iLastDiff), ffNumber, 10, 2) _
                                & " Tcmp=" & Str$(iDuration) & "ms " & sTmp & sBeep )
    iLastDiff = iDiff
End sub

Sub OnKeypress_frmDetectMotion (key As Byte)
    DefDbl dblThresh
    If (key = 43 or key = 45) Then
       If key = 43 Then Inc iThreshold
       If key = 45 Then Dec iThreshold
       If iThreshold < 0 Then iThreshold = 0
       dblThresh = iThreshold/10
       lblThreshold.Caption = " THRESHOLD : " & Str$(dblThresh) & " pc (+/- change) "
    End If
End Sub

Sub OnClic_AnyMenu (Sender)
    If flagMenuBusy = True Then Exit Sub
    flagMenuBusy = true
    Select Case Sender.Handle
      Case mnuFormatDlg.Handle
          If hCamWnd = 0 Then rchWin.additems (" *** ERROR *** : CAM MUST BE OPENED TO SET VIDEO FORMAT"):Exit Sub
          tmrSpy.Enabled = False
          rchWin.additems ("VIDEO FORMAT DIALOG  = " & Str$(SendMessageX (hCamWnd, WM_CAP_DLG_VIDEOFORMAT, 0,
0)) & "  (If 0, You Have None)")
          tmrSpy.Enabled = True
       Case mnuSourceDlg.Handle
          If hCamWnd = 0 Then rchWin.additems (" *** ERROR *** : CAM MUST BE OPENED TO SET VIDEO SOURCE"):Exit Sub
          tmrSpy.Enabled = False
          rchWin.additems ("VIDEO FORMAT SOURCE  = " & Str$(SendMessageX (hCamWnd, WM_CAP_DLG_VIDEOSOURCE, 0,
0)) & "  (If 0, You Have None)")
          tmrSpy.Enabled = True
    End select
    flagMenuBusy = False
End Sub
```

24.5 QWEBCAM

This element is a WebCam controller whereby you are able to set the size as well. With the previous element (direct show), the user had to set the size. It is also much more stable, fast and free memory as it works, unlike the previous method again that uses a lot of memory. The coding is also much less and less complicated. You will however have to include the WEBCAM.DLL, WEBCAM.LIB and VFW.INC

SNO	CODING	EXPLANATION
1	$TYPECHECK ON $INCLUDE "RapidQ2.inc" $INCLUDE "vfw.inc"	Include the element VFW
	DECLARE SUB CleanUp_WebCam DECLARE SUB StartTheCamUp declare sub shownow	Sub programs to activate and call the web cam.
	CREATE CamForm AS QFORM BorderStyle = bsToolWindow Caption = "vfw CAM":Center OnClose = CleanUp_WebCam onshow=shownow END CREATE	Display when FORM is displayed.
	CREATE MyCam AS QWebCam left=0:top=0 ShowErrors = True BitCount = 24 Capture =true END CREATE CamForm.ShowModal	Establish the element.
	sub shownow StartTheCamUp end sub	Start the camera.

	SUB StartTheCamUp
	IF MyCam.CamInit(CamForm) THEN
	CamForm.Show
	MyCam.Preview = True
	IF MyCam.SetImageSize(320, 240) = False THEN showmessage "cannot set size"
	MyCam.Resize(CamForm)
	ELSE
	Showmessage "Error: cannot initialize Camera"
	Application.Terminate
	END IF
	END SUB
		Start the camera now.
		Preview on.
		Set size (160/120, 320/240, 352/288, 640/480)

	SUB CleanUp_WebCam MyCam.CleanUp IF CamForm.Visible THEN CamForm.Close END SUB	Closes the webcam and clears all memory usage.
2	$TYPECHECK ON $INCLUDE "RapidQ2.inc" $INCLUDE "vfw.inc" DECLARE SUB CleanUp_WebCam DECLARE SUB StartTheCamUp declare sub shownow declare sub setsize declare sub setoption declare sub setdisplay declare sub setcompression	All Settings

SNO	CODING	EXPLANATION

	CREATE CamForm AS QFORM
	BorderStyle = bsToolWindow
	Caption = "vfw CAM":Center
	OnClose = CleanUp_WebCam
	OnClick = StartTheCamUp
	onshow=shownow
	create menu as qmainmenu
	create menua as qmenuitem
	caption="Size":onclick=setsize
	end create
	create menub as qmenuitem
	caption="Options":onclick=setoption
	end create
	create menuc as qmenuitem
	caption="Display":onclick=setdisplay
	end create
	create menud as qmenuitem
	caption="Compression":onclick=setcompression
	end create
	end create
	END CREATE
	CREATE MyCam AS QWebCam
	left=0:top=0:ShowErrors = True
	BitCount = 24:Capture =true
	END CREATE
	CamForm.ShowModal
	
	sub shownow
	StartTheCamUp
	MyCam.Resize(CamForm)
	end sub
	
	SUB StartTheCamUp
	IF MyCam.CamInit(CamForm) THEN
	CamForm.Show
	MyCam.Preview = True
	IF MyCam.SetImageSize(320, 240) = False THEN showmessage "cannot set size"
	MyCam.Resize(CamForm)
	ELSE
	Showmessage "Error: cannot initialize Camera"
	Application.Terminate
	END IF
	END SUB
	
	SUB CleanUp_WebCam
	MyCam.CleanUp
	IF CamForm.Visible THEN CamForm.Close
	END SUB
	
	SUB setsize
	MyCam.ShowDialogBox(1)
	MyCam.GetStatus
	IF MyCam.SetImageSize(MyCam.ImageWidth, MyCam.ImageHeight) = False THEN
	showmessage "cannot set size"
	ELSE
	MyCam.Resize(CamForm)
	END IF
	END SUB
	
	SUB setoption
	MyCam.ShowDialogBox(2)
	MyCam.GetStatus:MyCam.Resize(CamForm)
	END SUB
	
	SUB setdisplay
	MyCam.ShowDialogBox(3)
	MyCam.GetStatus
	MyCam.Resize(CamForm)
	END SUB	

SNO	CODING	EXPLANATION
	SUB setcompression MyCam.ShowDialogBox(4) MyCam.GetStatus MyCam.Resize(CamForm) END SUB	

3	$TYPECHECK ON
	$INCLUDE "RapidQ2.inc"
	$INCLUDE "vfw.inc"
	
	DECLARE SUB CleanUp_WebCam
	DECLARE SUB StartTheCamUp
	declare sub shownow
	declare sub setsize
	declare sub setoption
	declare sub setdisplay
	declare sub setcompression
	declare sub takephoto
	
	CREATE CamForm AS QFORM
	BorderStyle = bsToolWindow:Caption = "vfw CAM":Center
	OnClose = CleanUp_WebCam:OnClick = StartTheCamUp
	onshow=shownow
	create menu as qmainmenu
	create menua as qmenuitem
	caption="Size":onclick=setsize
	end create
	create menub as qmenuitem
	caption="Options":onclick=setoption
	end create
	create menuc as qmenuitem
	caption="Display":onclick=setdisplay
	end create
	create menud as qmenuitem
	caption="Compression":onclick=setcompression
	end create
	create menue as qmenuitem
	caption="Photo":onclick=takephoto
	end create
	end create
	END CREATE
	
	CREATE MyCam AS QWebCam
	left=0:top=0:ShowErrors = True
	BitCount = 24:Capture =true
	END CREATE
	CamForm.ShowModal
	
	sub shownow
	StartTheCamUp
	MyCam.Resize(CamForm)
	end sub
	
	SUB StartTheCamUp
	IF MyCam.CamInit(CamForm) THEN
	CamForm.Show
	MyCam.Preview = True
	IF MyCam.SetImageSize(320, 240) = False THEN showmessage "cannot set size"
	MyCam.Resize(CamForm)
	ELSE
	Showmessage "Error: cannot initialize Camera"
	Application.Terminate
	END IF
	END SUB
	
	SUB CleanUp_WebCam
	MyCam.CleanUp
	IF CamForm.Visible THEN CamForm.Close
	END SUB
		Taking a photo

SNO	CODING	EXPLANATION

	SUB setsize
	MyCam.ShowDialogBox(1)
	MyCam.GetStatus
	IF MyCam.SetImageSize(MyCam.ImageWidth, MyCam.ImageHeight) = False THEN
	showmessage "cannot set size"
	ELSE
	
	MyCam.Resize(CamForm)
	END IF
	END SUB
	
	SUB setoption
	MyCam.ShowDialogBox(2)
	MyCam.GetStatus:MyCam.Resize(CamForm)
	END SUB
	
	SUB setdisplay
	MyCam.ShowDialogBox(3)
	MyCam.GetStatus:MyCam.Resize(CamForm)
	END SUB
	
	SUB setcompression
	MyCam.ShowDialogBox(4)
	MyCam.GetStatus:MyCam.Resize(CamForm)
	END SUB
	
	sub takephoto
	MyCam.SaveFrameToFile("c:\ms3\~common\testpht.bmp")
	end sub	