S.T.R.I.K.E.7 API Specification

Specification v2.2

Contents

Connection lifecycle

- *Client code loads the API DLL.*
- *DLL Setup. Registering for events.*
- *The Client code registers itself with the STRIKE API (OpenApplet). This includes a Name and Icon. The call will return a unique identifier for this Applet. More than one applet can be registered with the API.*
- *The client code then waits for an AppletActive event. This will include the applet Id. At this point the client code can start sending images.*
- *The applet will receive touch screen events.*
- *When the user presses the home button on the S.T.R.I.K.E.7 the applet will receive AppletInactive (Paused) event.*
- *If the Client program closes it must call the CloseApplet function.*

Events (Callbacks)

- `typedef void (*AppletActiveEvent)( uint16_t appletId, uint16_t endPoindId, uint32_t endPointType );`

- `typedef void (*AppletInactiveEvent)( uint16_t appletId, uint16_t endPointId );`

- `typedef void (*AppletTouchEvent)( uint16_t appletId, uint16_t endPointId, uint32_t x, uint32_t y, BOOL pressed );`

- `typedef void (*AppletResetEvent)( uint16_t appletId, uint16_t endPointId );`

AppletActiveEvent

Definition

```
Typedef void (*AppletActiveEvent( uint16_t appletId, uint32_t endPointType, uint16_t
endPoindId );
```

Parameters

appletId – This is the unique identifier which is used to identify an applet. The appletId is auto generated and passed back from the API via the OpenApplet function call.

endPointType – This identifies what type of endpoint has made your applet active. This is the VID / PID of the device concatenated together so for example VID: 0x0738 PID: 0x1234 would equate to and end point type of 0x07381234.

endPointId – This this the same as appletId but for end points. End point Id 0 (Zero) is special and means broadcast to all endpoints.

Description

After an applet has been opened with the OpenApplet function it must wait for the user to activate it, by selecting to view it via the V.E.N.O.M. screen. This event tells the applet that it has been selected and that it can now start rendering to the V.E.N.O.M. screen. This event is sent every time the user chooses to view the applet.

AppletInactiveEvent

Definition

```
typedef void (*AppletInactiveEvent)( uint16_t appletId, uint16_t endPoindId );
```

Parameters

appletId – This is the unique identifier which is used to identify an applet. The appletId is auto generated and passed the back from the API via the OpenApplet function call.

endPointId – This this the same as appletId but for end points. End point Id 0 (Zero) is special and means broadcast to all endpoints.

Description

At any time a user can navigate away from an applet by pressing the home button on the S.T.R.I.K.E.7 command module. AppletInactiveEvent will be sent to the applet to let it know that it is no longer visible to the user. The applet should stop trying to render to the V.E.N.O.M. screen (all render requests will be ignored). The user may choose to view the applet again at some point, so this can be treated like a 'pause' event.

AppletTouchEvent

Definition

```
typedef void (*AppletTouchEvent)( uint16_t appletId, uint16_t endPoindId, uint32_t x,
uint32_t y, BOOL pressed );
```

Parameters

appletId – This is the unique identifier which is used to identify an applet. The appletId is auto generated and passed the back from the API via the OpenApplet function call.

endPointId – This this the same as appletId but for end points. End point Id 0 (Zero) is special and means broadcast to all endpoints.

x – The horizontal position of the touch event.

y – The vertical position of the touch event.

pressed – TRUE when the touch event refers to either a finger being placed on the screen or dragged across the screen. FALSE when the event refers to a finger being removed from the screen.

Description

Once an applet is running and active it will receive all touch events from the V.E.N.O.M. screen. This event will be triggered every time the user touches the V.E.N.O.M. screen, at regular intervals when the touch is dragged over the screen.

AppletResetEvent

Definition

```
typedef void (*AppletResetEvent)( uint16_t appletId, uint16_t endPoindId );
```

Parameters

appletId – This is the unique identifier which is used to identify an applet. The appletId is auto generated and passed the back from the API via the OpenApplet function call.

endPointId – This this the same as appletId but for end points. End point Id 0 (Zero) is special and means broadcast to all endpoints.

Description

If the user unplugs the S.T.R.I.K.E.7 keyboard from the system. The API will send this event for any applets that were currently running on the V.E.N.O.M. screen. This includes all applets that have called the OpenApplet function, even if they have not been active. This event can be ignored if the developer would like their Applet to keep state between S.T.R.I.K.E.7 disconnects and reconnects.

Functions (Entry points)

- `void RegisterAppletActiveEvent( AppletActiveEvent * func, uint16_t appletId = 0 );`
- `void RegisterAppletInactiveEvent( AppletInactiveEvent * func, uint16_t appletId = 0 );`
- `void RegisterAppletTouchEvent( AppletTouchEvent * func, uint16_t appletId = 0 ) ;`
- `void RegisterAppletResetEvent(AppletResetEvent * func, uint16_t appletId = 0 );`
- `uint16_t OpenApplet( const WCHAR * name, const void * iconData, uint32_t * supportedDevices );`
- `void CloseApplet( uint16_t appletId, uint16_t endPointId );`
- `void AppletUpdateScreen( uint16_t appletId, uint16_t endPointId );`
- `void AppletScroll( uint16_t appletId, uint16_t endPointId, uint32_t fromX, uint32_t fromY, uint32_t width, uint32_t height, int32_t distanceX, int32_t distanceY );`
- `void AppletDraw( uint16_t appletId, uint16_t endPointId, int32_t x, int32_t y, uint32_t width, uint32_t height, const void * data );`
- `void AppletFill( uint16_t appletId, uint16_t endPointId, int32_t x, int32_t y, uint32_t width, uint32_t height, uint32_t colour );`
- `void AppletDrawText( uint16_t appletId, uint16_t endPointId, int32_t x, int32_t y, uint32_t size, uint32_t colour, WCHAR * text );`

NOTE: All GFX functions operate on an internal buffer. To make the screen updates visible you must call AppletUpdateScreen() after you have finished applying your GFX updates.

RegisterAppletActiveEvent

Definition

```
void RegisterAppletActiveEvent(AppletActiveEvent * func, uint16_t appletId = 0 );
```

Parameters

func – A pointer to the callback function that is being registered.

appletId – The id of the applet that the event is being registered for. AppletId = 0 is the default and means all applets.

Description

Registers the function that will be called when the AppletActive event is fired.

RegisterAppletInactiveEvent

Definition

```
void RegisterAppletInactiveEvent( AppletInactiveEvent * func, uint16_t appletId = 0 );
```

Parameters

func – A pointer to the callback function that is being registered.

appletId – The id of the applet that the event is being registered for. AppletId = 0 is the default and means all applets.

Description

Registers the function that will be called when the AppletInactive event is fired.

RegisterAppletTouchEvent

Definition

```
void RegisterAppletTouchEvent( AppletTouchEvent * func, uint16_t appletId = 0 );
```

Parameters

func – A pointer to the callback function that is being registered.

appletId – The id of the applet that the event is being registered for. AppletId = 0 is the default and means all applets.

Description

Registers the function that will be called when the AppletTouch event is fired.

RegisterAppletResetEvent

Definition

```
void RegisterAppletResetEvent( AppletResetEvent * func, uint16_t id = 0 );
```

Parameters

func – A pointer to the callback function that is being registered.

appletId – The id of the applet that the event is being registered for. AppletId = 0 is the default and means all applets.

Description

Registers the function that will be called when the AppletReset event is fired.

OpenApplet

Definition

```
uint16_t OpenApplet( const WCHAR* name, uint32_t iconWidth, uint32_t iconHeight, const
void * iconData, uint32_t * supportedDevices );
```

Parameters

name – The name of the applet that is being opened.

iconWidth – The width of the icon.

iconHeight – The height of the icon.

iconData – Raw bitmap data for the icon. The data is an array of iconHeight x iconWidth x 4 bytes, with each pixel taking up four bytes in the BGRA order. Pixels are grouped in rows from left to right, and rows are grouped from top to bottom.

supportedDevices – An array of device ids that this applet supports. A device ID is the combination of a 16bits VID (in the most significant half) and a 16 bits PID (in the least significant half). This allows applets to restrict the devices it was developed for. The array should be terminated with a 0 entry. If the array is empty, or NULL is passed, all devices are supported.

Description

This tells the API that an applet is now open. The function call includes an icon to display on the V.E.N.O.M. screen.

Supported Device Ids

- *STRIKE7: 0x0738a109*

CloseApplet

Definition

```
void CloseApplet( uint16_t appletId, uint16_t endPointId );
```

Parameters

appletId – The id of the applet that is being closed.

endPointId – This this the same as appletId but for end points. End point Id 0 (Zero) is special and means broadcast to all endpoints.

Description

Tells the API that the applet is now closed. The applet icon will be removed from the V.E.N.O.M. screen.

AppletUpdateScreen

Definition

```
void AppletUpdateScreen( uint16_t appletId, uint16_t endPointId );
```

Parameters

appletId – The id of the applet that this command is for.

endPointId – This this the same as appletId but for end points. End point Id 0 (Zero) is special and means broadcast to all endpoints.

Description

Updates the screen with the latest GFX operations.

To allow for seamless screen updates, all GFX functions will not update the V.E.N.O.M. screen until this function is called. This means that a programmer can perform several GFX operations before the V.E.N.O.M. screen is updated. This is essentially double buffering.

AppletScroll

Definition

```
void AppletScroll( uint16_t appletId, uint16_t endPointId, uint32_t fromX, uint32_t
fromY, uint32_t width, uint32_t height, int32_t distanceX, int32_t distanceY );
```

Parameters

appletId – The id of the applet that this command is for.

endPointId – This this the same as appletId but for end points. End point Id 0 (Zero) is special and means broadcast to all endpoints.

fromX – The position in the X-Axis where to start scrolling from.

fromY – The position in the Y-Axis where to start scrolling from.

width – The width of the rectangle to scroll.

height –The height of the rectangle to scroll.

distanceX – The number of pixels to shift the selection along the X-Axis.

distanceY – The number of pixels to shift the selection along the Y-Axis.

Description

Scrolls a rectangular section of the screen to another location on the V.E.N.O.M. screen.

NOTE: This is not a move function. The scroll happens within the specified rectangle.

AppletDraw

Definition

```
void AppletDraw( uint16_t appletId, uint16_t endPointId, int32_t x, int32_t y,
uint32_t width, uint32_t height, const void * data );
```

Parameters

appletId – The id of the applet

endPointId – This this the same as appletId but for end points. End point Id 0 (Zero) is special and means broadcast to all endpoints.

x – Where to render the image horizontally on the V.E.N.O.M. screen.

y – Where to render the image vertically on the V.E.N.O.M. screen.

width – The width of the image being sent.

height – The height of the image being sent.

data – Raw bitmap data for the image that is being sent to the V.E.N.O.M. screen. The data is an array of height x width x 4 bytes, with each pixel taking up four bytes in the BGRA order. Pixels are grouped in rows from left to right, and rows are grouped from top to bottom.

Description

Draws a piece of GFX at a specific location on the V.E.N.O.M. screen.

AppletFill

Definition

```
void Fill( uint16_t appletId, uint16_t endPointId, int32_t x, int32_t y, uint32_t
width, uint32_t height, uint32_t colour );
```

Parameters

appletId – The id of the applet

endPointId – This this the same as appletId but for end points. End point Id 0 (Zero) is special and means broadcast to all endpoints.

x – Left point of the fill rectangle.

y – Top point of the fill rectangle.

width – The width of the fill rectangle.

height – The height of the fill rectangle.

colour – The colour of the fill. Using ARGB colour space (red = 0xffff0000, green = 0xff00ff00, blue = 0xff0000ff).

Description

Fills in the section of the screen specified by the parameters with the stated colour.

AppletDrawText

Definition

```
void AppletDrawText( uint16_t appletId, uint16_t endPointId, int32_t x, int32_t y,
uint32_t size, uint32_t colour, const WCHAR * text);
```

Parameters

appletId – The id of the applet

endPointId – This this the same as appletId but for end points. End point Id 0 (Zero) is special and means broadcast to all endpoints.

x – Top point of where to render the text

y – Left point of where to render the text.

size – The size of the text in points.

colour – The colour of the fill. Using ARGB colour space (red = 0xffff0000, green = 0xff00ff00, blue = 0xff0000ff).

text – The text to render.

Description

Provides a quick and easy way to output some text on to the V.E.N.O.M. screen.