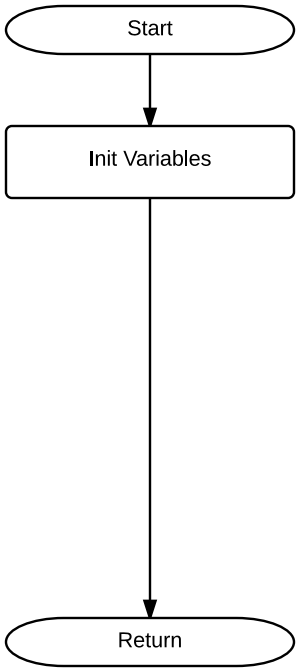
```
#define  TicksPerStep      0x04     ; it takes 4 transitions, detent-to-detent as a single "step"

SavedSwitchPos     RES    1    ; holds last, rotated switch position
SwitchValue        RES    1    ; this holds the number of counts of rotation 256 values
UpPrescale         RES    1    ; it takes 4 transitions, detent-to-detent as a single "step"
DownPrescale       RES    1    ; it takes 4 transitions, detent-to-detent as a single "step"

extern    Port_Old, Port_Changed
```



Code

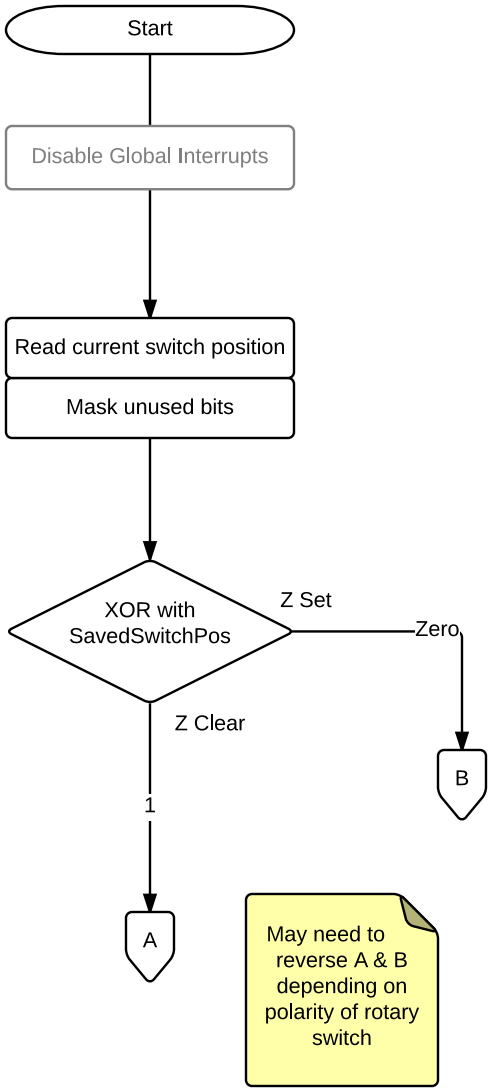
InitReadEncoder

```
CLRF    SavedSwitchPos
CLRF    SwitchValue
MOVLW   TicksPerStep
MOVWF   DownPrescale
MOVWF   UpPrescale
```

RETURN

Notes:
1) FUTURE: Will need to save/retrieve SwitchValue from EEPROM
2) FUTURE: Disable/Enable Interrupts

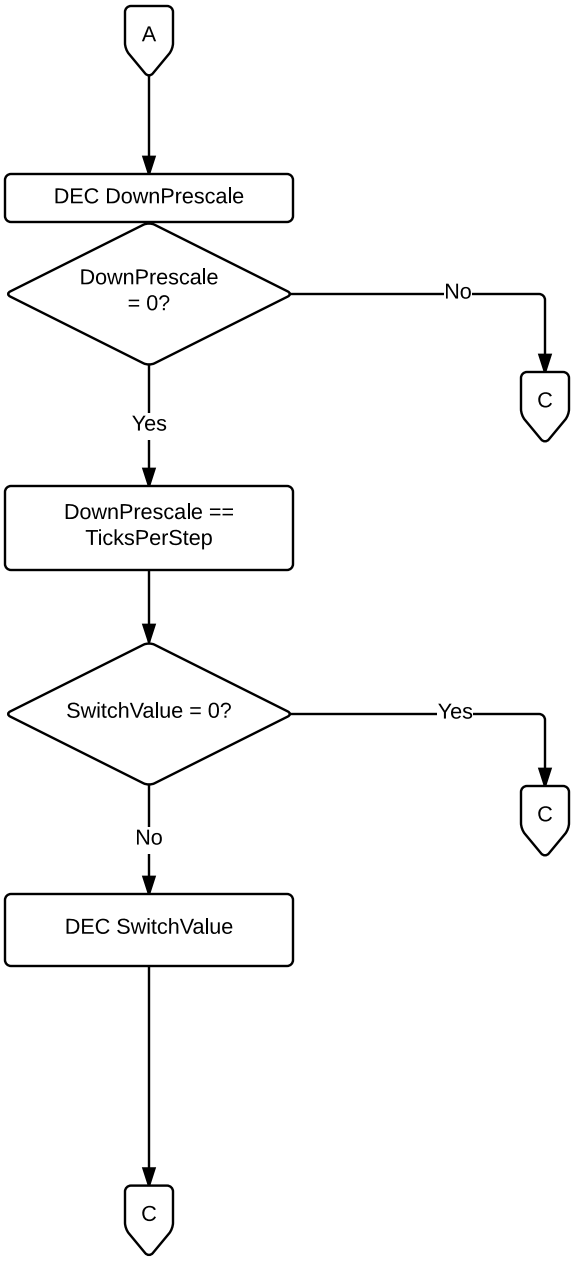
```
#define  TicksPerStep      0x04     ; it takes 4 transitions, detent-to-detent as a single "step"

SavedSwitchPos     RES    1    ; holds last, rotated switch position
SwitchValue        RES    1    ; this holds the number of counts of rotation 256 values
UpPrescale         RES    1    ; it takes 4 transitions, detent-to-detent as a single "step"
DownPrescale       RES    1    ; it takes 4 transitions, detent-to-detent as a single "step"

extern    Port_Old, Port_Changed
```



Code

```
; Called only when a CHANGE is detected in the
; switch position

ReadEncoder

    mDisableGI   ; Macro

    MOVLW B'00000001'
    ANDWF Port_Old, w

    XORWF SavedSwitchPos, w
    BTFSC   STATUS, Z

    GOTO SwitchUp

    GOTO SwitchDown
```

```
#define  TicksPerStep      0x04     ; it takes 4 transitions, detent-to-detent as a single "step"

SavedSwitchPos     RES    1    ; holds last, rotated switch position
SwitchValue        RES    1    ; this holds the number of counts of rotation 256 values
UpPrescale         RES    1    ; it takes 4 transitions, detent-to-detent as a single "step"
DownPrescale       RES    1    ; it takes 4 transitions, detent-to-detent as a single "step"

extern    Port_Old, Port_Changed
```

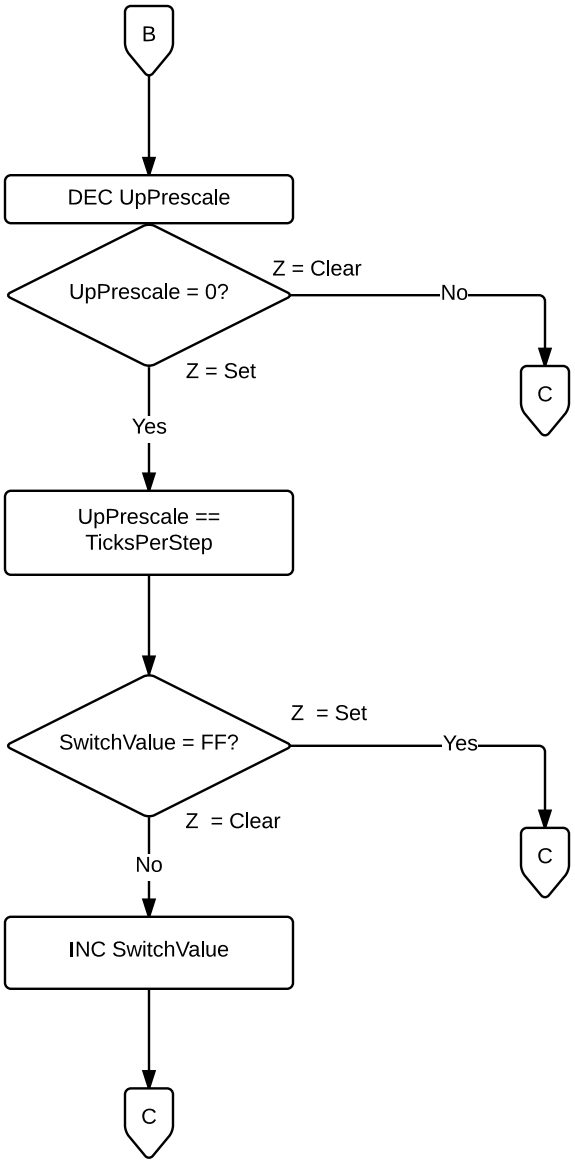


Code

SwitchDown

```
DECFSZ      DownPrescale, f

GOTO    RotateSave

MOVLW TicksPerStep
MOVWF DownPrescale

MOVF    SwitchValue, f
BTFSC   STATUS, Z

GOTO    RotateSave

DEC SwitchValue, f

GOTO RotateSave
```

```
#define  TicksPerStep      0x04     ; it takes 4 transitions, detent-to-detent as a single "step"

SavedSwitchPos     RES    1    ; holds last, rotated switch position
SwitchValue        RES    1    ; this holds the number of counts of rotation 256 values
UpPrescale         RES    1    ; it takes 4 transitions, detent-to-detent as a single "step"
DownPrescale       RES    1    ; it takes 4 transitions, detent-to-detent as a single "step"

extern    Port_Old, Port_Changed
```

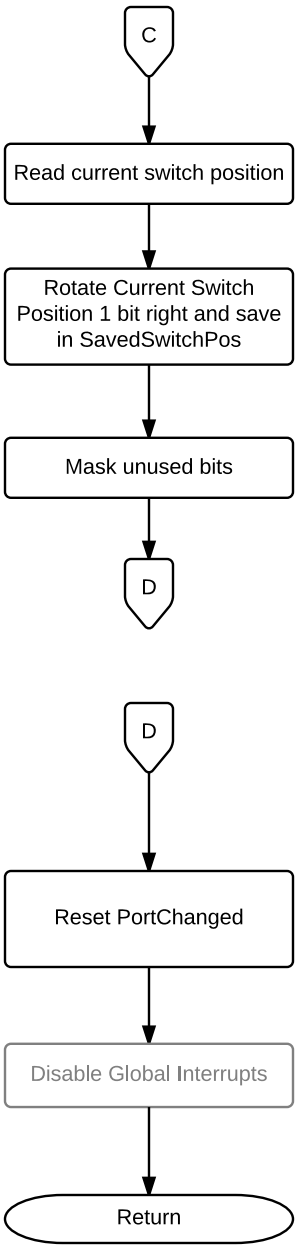


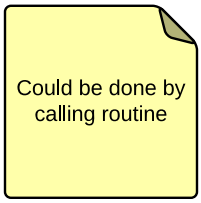
Code

SwitchUp

```
DECFSZ      UpPrescale, f

GOTO    RotateSave

MOVLW TicksPerStep
MOVWF UpPrescale

MOVLW 0xFF
XORWF SwitchValue, w
BTFSC   STATUS, Z

GOTO    RotateSave

GOTO RotateSave
```

```
#define  TicksPerStep      0x04     ; it takes 4 transitions, detent-to-detent as a single "step"

SavedSwitchPos     RES    1    ; holds last, rotated switch position
SwitchValue        RES    1    ; this holds the number of counts of rotation 256 values
UpPrescale         RES    1    ; it takes 4 transitions, detent-to-detent as a single "step"
DownPrescale       RES    1    ; it takes 4 transitions, detent-to-detent as a single "step"

extern    Port_Old, Port_Changed
```





Code

```
RotateSave

        MOVFF  Port_Old, SavedSwitchPos

        RRNCF  SavedSwitchPos, f

        MOVLW B'00000001'
        ANDWF  SavedSwitchPos, f

        GOTO    ExitReadEncoder

ExitReadEncoder

        MOVLW B'11111100'
        ANDWF  Port_Changed, f

        mDisableGI   ; Macro

        RETURN
```

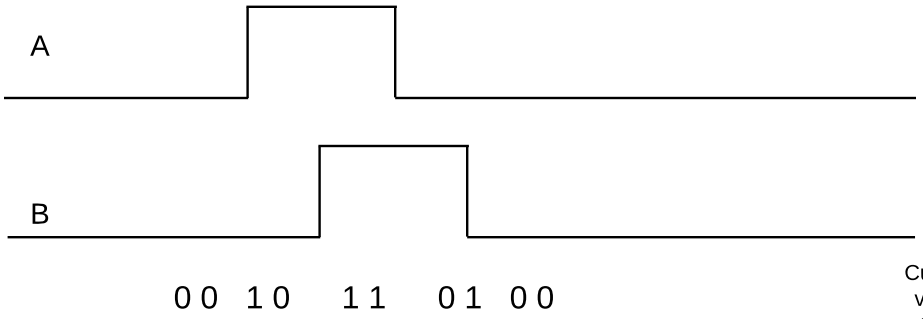
Revision History

02 Replaced single prescaler (SwitchCount) with two prescalers
UpPrescale and DownPrescale.

01 Initial Version

Determining rotation direction

Clockwise

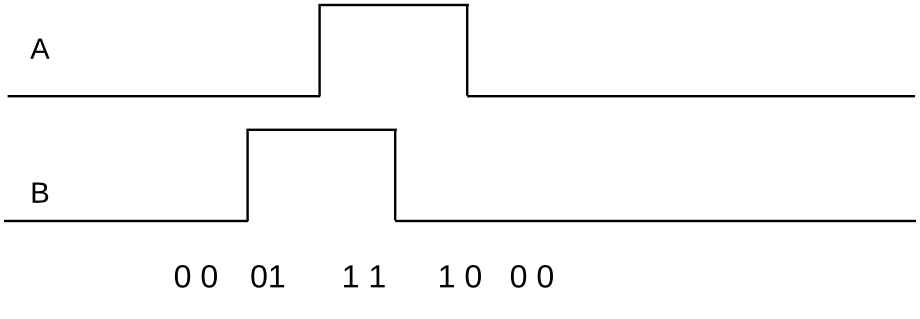


Blue arrow =
Rotate current value 1 bit to the right and save

Red arrow =
XOR rightmost bits only

Current value	Rotated, saved value	XOR value
0 0		
1 0	x 0	0
1 1	x 1	0
0 1	x 1	0
0 0	x 0	0

Counter Clockwise



Rotated, saved value	XOR value
0 0	
0 1	x 0
1 1	x 0
1 0	x 1
0 0	x 1