

Configuring the M-986 MF Trunk Signaling

Configuring the M-986

The following is an example of pseudocode of the commands required to control the M-986-2R2. Pseudocode for M-986 configured for external clocking, manual mode, binary operation, with end of digit operation.

```

BEGIN_MAIN_INIT:                                     // initialize the M-986

WAIT_LOOP1:
If( /RBLE is low)                                  // if the write buffer is empty
Output configuration byte one, 00110000B;          // initialize, channel 1 disabled
Else
  goto WAIT_LOOP1;                                  // go wait for write buffer to be empty
Endif

WAIT_LOOP2:
If( /RBLE is low)                                  // if the write buffer is empty
  Output configuration byte two, 01100000B;        // initialize with channel 2 disabled
Else
  goto WAIT_LOOP2;                                  // go wait for write buffer to be empty
Endif
Set config1 = 0;                                    // set no configuration byte 1 update required
Set config2 = 0;                                    // set no configuration byte 2 update required
Set ch1_state = IDLE;                               // set the state for the ch1 processing loop to idle
Set ch2_state = IDLE;                               // set the state for the ch2 processing loop to idle
Set ch1_tone = 0;                                   // clear ch 1 receive digit buffer
Set sav1_tone = 0;                                  // clear ch 1 saved digit
Set tone1 = 0;                                      // clear flag for ch1 digit received
Set ch2_tone = 0;                                   // clear ch 2 receive digit buffer
Set sav2_tone = 0;                                  // clear ch 2 saved digit
Set tone2 = 0;                                      // clear flag for ch2 digit received
Goto BEGIN_MAIN_LOOP                                // initialize is complete, go to main loop
END_MAIN_INIT:
BEGIN_MAIN_LOOP:
If( /TBLF transitions low )                         // if digit is received
  CPU drives /RD low to read the tone, 0S00XXXX where "S" is channel select bit and "XXXX" is the
  binary value.
  If( bit 6 = 0)                                    // bit 6 gives channel number
    ch1_tone = XXXX;                                // save value for later processing
    tone1 = 1;                                       // flag channel 1 tone received
  Else                                              // it is for channel 2
    ch2_tone = XXXX;                                // save value for later processing
  
```

```

    tone2 = 1; // flag channel 2 tone received
  Endif
Endif
If( config1 = 1) // if load of configuration byte one required
  If( /RBLE is low) // if transmit buffer is empty
    config1 = 0; // clear flag
    Output configuration byte config_one; // write configuration byte
  Endif
Endif
Else If( config2 = 1) // if load of configuration byte two required
  If( /RBLE is low) // if transmit buffer is empty
    config2 = 0; // clear flag
    Output configuration byte config_two; // write configuration byte
  Endif
Endif
Else If( ch1_trans = 1) // if transmit to channel 1 required
  If( /RBLE is low) // if transmit buffer is empty
    ch1_trans = 0; // clear flag
    Output transmit command, 1000YYYY // "YYYY" is binary value
  Endif
Endif
Else If( ch2_trans = 1) // if transmit to channel 2 required
  If( /RBLE is low) // if transmit buffer is empty
    ch2_trans = 0; // clear flag
    Output transmit command, 1100YYYY // "YYYY" is binary value
  Endif
Endif
Call PROCESS_CH1; // do processing for channel 1
Call PROCESS_CH2; // do processing for channel 2
Goto BEGIN_MAIN_LOOP; // go back to start of main loop
END_MAIN_LOOP:
BEGIN_PROCESS_CH1:
PROCESS_CH1:
  If(ch1_state = RESET) // if reset state
    If( ch1_trans = 0) // if no transmit command pending
      config1 = 1; // signal configuration byte 1 update required
      config_one = 00110100 B (binary); // save configuration to be loaded
      // which is channel disable
    Set ch1_state = IDLE; // switch to idle on next pass
  Endif;
Endif;

```

```

Else If(ch1_state = IDLE)                                // idle, wait for assignment to central office
  If(channel 1 assigned for backward operation) // if channel assigned to central office
    config1 = 1;                                         // signal configuration byte 1 update required
    config_one = 00111100 B (binary); // save configuration to be loaded
                                                    // channel enabled in backward mode
    Set ch1_state = BACKWARD;                            // switch to backward state on next pass
  Endif;
  If(channel 1 assigned for forward operation)           // if channel assigned to central office
    config1 = 1;                                         // signal configuration byte 1 update required
    config_one = 00111101 B (binary);                   // save configuration to be loaded
    tone1 = 1;                                          // set flag to start forward transmit
    ch1_tone = 0;                                       // also insures start of forward digit
    Set ch1_state = FORWARD;                            // switch to forward state on next pass
  Endif;
Endif;
Else If(ch1_state = BACKWARD)                           // backward, wait for forward digit & send reply
  If( tone1 = 1)                                       // if channel 1 tone seen
    tone1 = 0;                                         // clear the flag
    If(ch1_tone = 0)                                    // if value received shows end of digit
      ch1_trans = 1;                                    // flag to send command to transmitter
      trans1 = 0000;                                    // zero tone value to turn tone off
      If( sav1_tone = last digit)                       // if last digit has been received
        Set ch1_state = Reset; // go to reset on next pass
      Endif;
    Else                                              // a non-zero digit value has been received
      sav1_tone = ch1_tone;                             // save for later checking of last digit
      ch1_trans = 1;                                    // flag to send command to transmitter
      trans1 = YYYY;                                    // YYYY is the acknowledge tone to transmit
    Endif;
  Endif;
Endif;
Else If(ch1_state = FORWARD)                           // forward, send forward digit & wait for reply
  If( tone1 = 1)                                       // if channel 1 backward tone seen
    tone1 = 0;                                         // clear the flag
    If(ch1_tone = 0)                                    // if value received shows end of digit
      If(last digit sent)                               // if this is the last digit in the call setup
        Set ch1_state = RESET;                         // back to reset on next pass
      Else
        ch1_trans = 1;                                  // flag to send command to transmitter
        trans1 = YYYY;                                  // YYYY is the forward tone to transmit
      Endif;
    Endif;
  Endif;
Endif;

```

```
    Endif;
  Else
    ch1_trans = 1;
    trans1 = 0000;
  Endif;
Endif;
Endif;
END_PROCESS_CH1:
BEGIN_PROCESS_CH2:
PROCESS_CH2:
  If(ch2_state = RESET)
    If( ch2_trans = 0)
      config2 = 1;
      config_two = 01100100 B (binary);

      Set ch2_state = IDLE;
    Endif;
  Endif;
Else If(ch2_state = IDLE)
  If(channel 2 assigned for backward operation)
    config2 = 1;
    config_two = 01101100 B (binary);

    Set ch2_state = BACKWARD;
  Endif;
  If(channel 2 assigned for forward operation)
    config2 = 1;
    config_two = 01101101 B (binary);
    tone2 = 1;
    ch2_tone = 0;
    Set ch2_state = FORWARD;
  Endif;
Endif;

Else If(ch2_state = BACKWARD)
  If( tone2 = 1)
    tone2 = 0;
    If(ch2_tone = 0)
      ch2_trans = 1;
      trans2 = 0000;
    Endif;
  Endif;
Endif;
```

```
// valid acknowledge has been received
// flag to send command to transmitter
// value to turn transmitter off
```

```
// if reset state
// if no transmit command pending
// signal configuration byte 1 update required
// save configuration to be loaded
// which is channel disable
// switch to idle no next pass
```

```
// idle, wait for assignment to central office
// if channel assigned to central office
// signal configuration byte 2 update required
// save configuration to be loaded
// channel enabled in backward mode
// switch to backward state on next pass
```

```
// if channel assigned to central office
// signal configuration byte 2 update required
// save configuration to be loaded
// set flag to start forward transmit
// also insures start of forward digit
// switch to forward state on next pass
```

```
// backward, wait for forward digit & send reply
// if channel 2 tone seen
// clear the flag
// if value received shows end of digit
// flag to send command to transmitter
// zero tone value to turn tone off
```

```
    If( sav2_tone = last digit)                // if last digit has been received
        Set ch2_state = RESET;                // go to reset on next pass
    Endif;
Else                                           // a non-zero digit value has been received
    sav2_tone = ch2_tone;                     // save for later checking of last digit
    ch2_trans = 1;                           // flag to send command to transmitter
    trans2 = YYYY;                           // YYYY is the acknowledge tone to transmit
Endif;
Endif;
Endif;
Else If(ch2_state = FORWARD)                 // forward, send forward digit & wait for reply
    If( tone2 = 1)                            // if channel 1 backward tone seen
        tone2 = 0;                          // clear the flag
    If(ch2_tone = 0)                          // if value received shows end of digit
        If(last digit sent)                  // if this is the last digit in the call setup
            Set ch2_state = RESET;          // back to RESET on next pass
    Else
        ch2_trans = 1;                       // flag to send command to transmitter
        trans2 = YYYY;                       // YYYY is the forward tone to transmit
    Endif;
    Else
        ch2_trans = 1;                       // flag to send command to transmitter
        trans2 = 0000;                       // zero tone value for off
    Endif;
Endif;
Endif;
END_PROCESS_CH2;
```

For additional information please visit our website at: www.ixysic.com

IXYS Integrated Circuits Division makes no representations or warranties with respect to the accuracy or completeness of the contents of this publication and reserves the right to make changes to specifications and product descriptions at any time without notice. Neither circuit patent licenses nor indemnity are expressed or implied. Except as set forth in IXYS Integrated Circuits Division's Standard Terms and Conditions of Sale, IXYS Integrated Circuits Division assumes no liability whatsoever, and disclaims any express or implied warranty, relating to its products including, but not limited to, the implied warranty of merchantability, fitness for a particular purpose, or infringement of any intellectual property right.

The products described in this document are not designed, intended, authorized or warranted for use as components in systems intended for surgical implant into the body, or in other applications intended to support or sustain life, or where malfunction of IXYS Integrated Circuits Division's product may result in direct physical harm, injury, or death to a person or severe property or environmental damage. IXYS Integrated Circuits Division reserves the right to discontinue or make changes to its products at any time without notice.

Specification: AN-125-R02
©Copyright 2014, IXYS Integrated Circuits Division
All rights reserved. Printed in USA.
4/10/2014