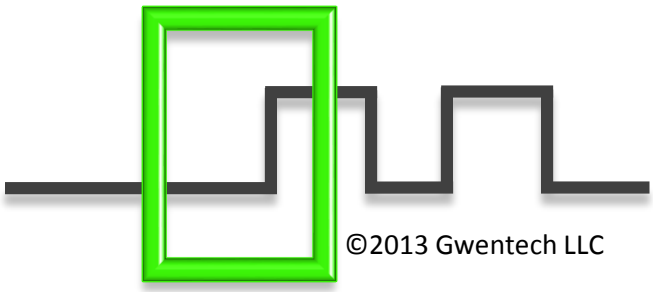
Receiving CAN Message to the Android® Application from the system:

buf[0]=1; // Packet type (message received is 1)
buf[1]=0; //Unused
buf[2]=0;// Unused
buf[3] // This packet length HIGH byte (0 always)
buf[4] // This packet length LOW byte (25 always, 5 byte header assumed)
buf[5] = ((id&0xFF000000)>>24);
buf[6] = ((id&0xFF0000)>>16);
buf[7] = ((id&0xFF00)>>8);
buf[8] = ((id&0xFF));
buf[9] = 0;//Unused
buf[10] = 0 ;//Unused
buf[11] = 0 ;//Unused
buf[12] = 0 ;//Unused
buf[13] = ((timer_value&0xFF)>>8);timer_value is between 0 and 7500, with 7500 being 1ms. (133.3 us)
buf[14] = (timer_value&0xFF);
buf[15] = ((ms_counter&0xFF000000)>>24); ms_counter between 0 and 0xFFFFFFFF
buf[16] = (ms_counter&0xFF0000)>>16;
buf[17] = ((ms_counter&0xFF00)>>8);
buf[18] = (ms_counter&0xFF);
buf[19] = ((day_counter&0xFF)>>8);
buf[20] = (day_counter&0xFF);day_counter between 0 and 999
buf[21]=appli_rx_msg.dlc;// Message length
buf[22]=appli_rx_msg.can_msg->data.s8[0];
buf[23]=appli_rx_msg.can_msg->data.s8[1];
buf[24]=appli_rx_msg.can_msg->data.s8[2];
buf[25]=appli_rx_msg.can_msg->data.s8[3];
buf[26]=appli_rx_msg.can_msg->data.s8[4];
buf[27]=appli_rx_msg.can_msg->data.s8[5];
buf[28]=appli_rx_msg.can_msg->data.s8[6];
buf[29]=appli_rx_msg.can_msg->data.s8[7];

Sending CAN Message from the Android® Application to the system:

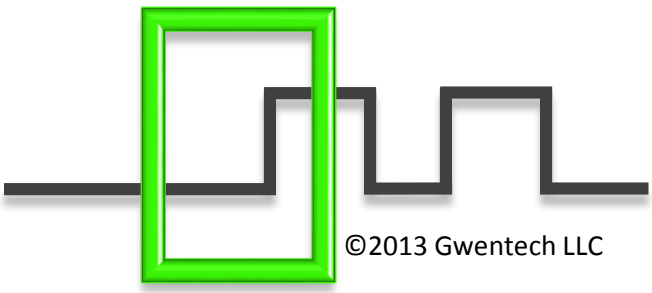
buf[0]=2; // Packet type (message send is 2)
buf[1]=0; //Unused
buf[2]=0;// Unused
buf[3] // This packet length HIGH byte (0 always)
buf[4] // This packet length LOW byte (13 always, 5 byte header assumed)
buf[5] = ((id&0xFF000000)>>24);
buf[6] = ((id&0xFF0000)>>16);
buf[7] = ((id&0xFF00)>>8);
buf[8] = ((id&0xFF));
buf[9]=appli_rx_msg.dlc;// Message length
Buf[10]=appli_tx_msg.can_msg->data.s8[0];
buf[11]=appli_tx_msg.can_msg->data.s8[1];
buf[12]=appli_tx_msg.can_msg->data.s8[2];
Buf[13]=appli_tx_msg.can_msg->data.s8[3];
buf[14]=appli_tx_msg.can_msg->data.s8[4];
buf[15]=appli_tx_msg.can_msg->data.s8[5];
buf[16]=appli_tx_msg.can_msg->data.s8[6];
buf[17]=appli_tx_msg.can_msg->data.s8[7];



Setting Baud Rate:

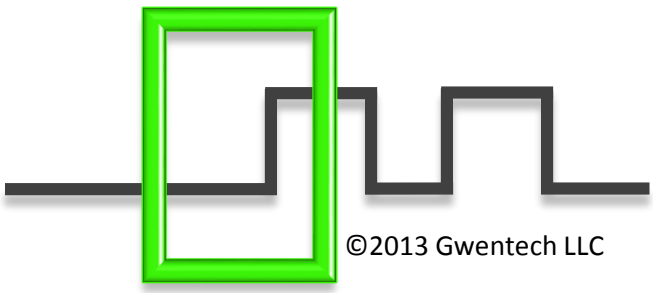
buf[0]=0x11; // Packet type (baud setting is 0x11)
buf[1]=0; //Unused
buf[2]=0;// Unused
buf[3]=0; // This packet length HIGH byte (0 always)
buf[4] =1;// This packet length LOW byte (1 always, 5 byte header assumed)
buf[5] = Baud setting

Baud setting is:
10 =125 Kbps
11 =250 Kbps
12 = 500 Kbps (Default)
13 = 800 Kbps
14 = 1 Mbps



Resetting time stamp:

buf[0]=0x10; // Packet type
buf[1]=0; //Unused
buf[2]=0;// Unused
buf[3]=0 // This packet length HIGH byte (0 always)
buf[4] =0// This packet length LOW byte (0 always, 5 byte header assumed)
buf[5] = 0//always zero



Read Device Serial Number:

buf[0]=0x12; // Packet type
buf[1]=0; //Unused
buf[2]=0;// Unused
buf[3]=0 // This packet length HIGH byte (0 always)
buf[4] =0// This packet length LOW byte (0 always, 5 byte header assumed)
buf[5] = 0//always zero

Device will reply with:

buf[0]=0x12; // Packet type
buf[1]=0; //Unused
buf[2]=0;// Unused
buf[3]=0 // This packet length HIGH byte (0 always)
buf[4] =15// This packet length LOW byte (15 always, 5 byte header assumed)
Buf[5] = ID Byte 0
Buf[6] = ID Byte 1
Buf[7] = ID Byte 2
Buf[8] = ID Byte 3
Buf[9] = ID Byte 4
Buf[10] = ID Byte 5
Buf[11] = ID Byte 6
Buf[12] = ID Byte 7
Buf[13] = ID Byte 8
Buf[14] = ID Byte 9
Buf[15] = ID Byte 10
Buf[16] = ID Byte 11
Buf[17] = ID Byte 12
Buf[18] = ID Byte 13
Buf[19] = ID Byte 14

Send Filter:

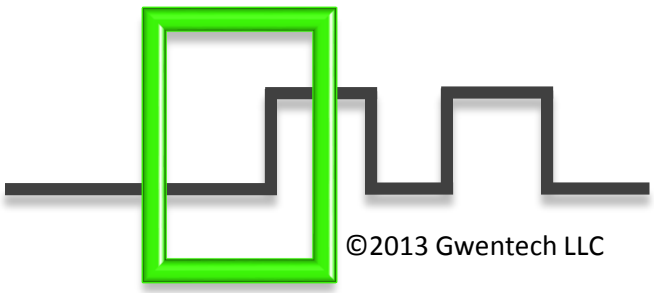
buf[0]=0x13; // Packet type
buf[1]=0; //Unused
buf[2]=0;// Unused
buf[3]=0 // This packet length HIGH byte (0 always)
buf[4] =33// This packet length LOW byte (0 always, 5 byte header assumed)
Buf[5] = ID Mask 0xYY000000
Buf[6] = ID Mask 0x00YY0000
Buf[7] = ID Mask 0x0000YY00
Buf[8] = ID Mask 0x000000YY
Buf[9] = ID Comparator 0xYY000000
Buf[10] = ID Comparator 0x00YY0000
Buf[11] = ID Comparator 0x0000YY00
Buf[12] = ID Comparator 0x000000YY
Buf[13] = ID Enabled (0=disabled, 1=enabled)
Buf[14] = Data Byte 0 Mask
Buf[15] = Data Byte 0 Comparator
Buf[16] = Data Byte 0 Enabled
Buf[17] = Data Byte 1 Mask
Buf[18] = Data Byte 1Comparator
Buf[19] = Data Byte 1 Enabled
Buf[20] = Data Byte 2Mask
Buf[21] = Data Byte 2 Comparator
Buf[22] = Data Byte 2 Enabled
Buf[23] = Data Byte 3 Mask
Buf[24] = Data Byte 3 Comparator
Buf[25] = Data Byte 3 Enabled
Buf[26] = Data Byte 4 Mask
Buf[27] = Data Byte 4 Comparator
Buf[28] = Data Byte 4 Enabled
Buf[29] = Data Byte 5Mask
Buf[30] = Data Byte 5 Comparator
Buf[31] = Data Byte 5Enabled
Buf[32] = Data Byte 6 Mask
Buf[33] = Data Byte 6 Comparator
Buf[34] = Data Byte 6 Enabled
Buf[35] = Data Byte 7 Mask
Buf[36] = Data Byte 7 Comparator
Buf[37] = Data Byte 7 Enabled

Read Filter from Device:

buf[0]=0x14; // Packet type
buf[1]=0; //Unused
buf[2]=0;// Unused
buf[3]=0 // This packet length HIGH byte (0 always)
buf[4] =0// This packet length LOW byte (0 always, 5 byte header assumed)
buf[5] = 0//always zero

Device will respond with

buf[0]=0x94; // Packet type
buf[1]=0; //Unused
buf[2]=0;// Unused
buf[3]=0 // This packet length HIGH byte (0 always)
buf[4] =33// This packet length LOW byte (0 always, 5 byte header assumed)
Buf[5] = ID Mask 0xYY000000
Buf[6] = ID Mask 0x00YY0000
Buf[7] = ID Mask 0x0000YY00
Buf[8] = ID Mask 0x000000YY
Buf[9] = ID Comparator 0xYY000000
Buf[10] = ID Comparator 0x00YY0000
Buf[11] = ID Comparator 0x0000YY00
Buf[12] = ID Comparator 0x000000YY
Buf[13] = ID Enabled (0=disabled, 1=enabled)
Buf[14] = Data Byte 0 Mask
Buf[15] = Data Byte 0 Comparator
Buf[16] = Data Byte 0 Enabled
Buf[17] = Data Byte 1 Mask
Buf[18] = Data Byte 1Comparator
Buf[19] = Data Byte 1 Enabled
Buf[20] = Data Byte 2Mask
Buf[21] = Data Byte 2 Comparator
Buf[22] = Data Byte 2 Enabled
Buf[23] = Data Byte 3 Mask
Buf[24] = Data Byte 3 Comparator
Buf[25] = Data Byte 3 Enabled
Buf[26] = Data Byte 4 Mask
Buf[27] = Data Byte 4 Comparator
Buf[28] = Data Byte 4 Enabled
Buf[29] = Data Byte 5Mask
Buf[30] = Data Byte 5 Comparator
Buf[31] = Data Byte 5Enabled
Buf[32] = Data Byte 6 Mask
Buf[33] = Data Byte 6 Comparator
Buf[34] = Data Byte 6 Enabled
Buf[35] = Data Byte 7 Mask
Buf[36] = Data Byte 7 Comparator
Buf[37] = Data Byte 7 Enabled



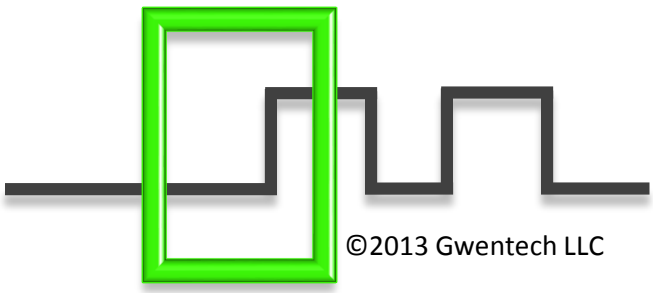
Read Firmware Version from Device:

buf[0]=0x15; // Packet type
buf[1]=0; //Unused
buf[2]=0;// Unused
buf[3]=0 // This packet length HIGH byte (0 always)
buf[4] =0// This packet length LOW byte (0 always, 5 byte header assumed)
buf[5] = 0//always zero

Device will respond with

buf[0]=0x95; // Packet type
buf[1]=0; //Unused
buf[2]=0;// Unused
buf[3]=0 // This packet length HIGH byte (0 always)
buf[4] =5// This packet length LOW byte (0 always, 5 byte header assumed)
Buf[5] = '1'
Buf[6] = '.'
Buf[7] = '0'
Buf[8] = '2'
Buf[9] = 0

Buf[5] through length, terminating with null will be the string of the firmware version. It will always be ASCII.



Additional Notes:

To use extended (29-bit) frames, set bit 29 (where ID bits are 0-28).

GT1026 Pinout on DB9 Connector:
Pin 1: No connection
Pin 2: CAN_L
Pin 3: CAN Ground
Pin 4: ---
Pin 5: CAN_H
Pin 6: CAN Ground
Pin 7: CAN_H
Pin 8: 120 ohm to CAN_L (Connect to CAN_H to include terminating resistor)
Pin 9: No Connection

GT1027 Pinout on DB9 Connector:
Pin 1: Vcc Ground
Pin 2: CAN_L
Pin 3: CAN Ground
Pin 4: ---
Pin 5: CAN_H
Pin 6: CAN Ground
Pin 7: CAN_H
Pin 8: 120 ohm to CAN_L (Connect to CAN_H to include terminating resistor)
Pin 9: +Vcc