

System 2X Checksum Application Note

Ian Fellows
LIMITED

If checksum is enabled a 16 Bit CRC Checksum will be appended to the weight data string. This parameter may be enabled either through the Serial Menu; CHSU=1, or through serial comms; EC=1. Further details can be found in the Operation Manual.

NOTE: Programming experience would be beneficial.

CRC Checksum

The CRC Checksum consists of: -

<STX><STRING><ETX><2 BYTES OF CHECKSUM><CARRIAGE RETURN>

Checksum Key

The last two bytes of any received string (after the **ETX** and before **CARRIAGE RETURN**) will be the **RECEIVED/SENT CHECKSUM**

Checking an incoming string

Extract the **CHECKSUM** (2 bytes long)

Put it in checksum_union to get an **UNSIGNED INT**:

```
csu.split.one=checksum_byte_1;//first  
csu.split.two=checksum_byte_2;//second
```

Then send it to be checked where:

buffer = received buffer

len_of_string = length of buffer up to and **NOT** including checksum bytes

```
if (!check_checksum(buffer,len_of_string,csu.check_sum))  
{  
  BAD CHECKSUM, here you generate error of some sorts  
}  
else  
  CHECKSUM OK
```

Creating an outgoing string (with checksum)

Send buffertosend (your outgoing string) to get_checksum, hence:

```
csu.check_sum=get_checksum(buffertosend);
```

Now that you have calculated the checksum append it to the end of your string (after the **ETX**)

Your first checksum byte to append will be csu.split.one

Your second checksum byte to append will be csu.split.two

Your outgoing string should have the following protocol:

<STX><STRING><ETX><2 BYTES OF CHECKSUM><CARRIAGE RETURN>

in other words

<STX><STRING><ETX><csu.split.one><csu.split.two><CARRIAGE RETURN>

```
union checksum_union
{
  struct
  {
    char two;//lower byte
    char one;//top byte
  }split;
  unsigned int check_sum;
}csu;
```

```
char check_checksum(char *buffer,int buflen,unsigned int check_sum)
{
  char val[80];

  memcpy(val,buffer,buflen);
  val[buflen]=0;
  if(check_sum!=get_checksum(val))
    return 0;//error
  else
    return 1;//ok
}
```

```
unsigned int get_checksum(char *buffer)
{
  unsigned int check_sum=0;

  while (*buffer)
    check_sum=updcrc(*buffer++,check_sum);
  return check_sum;
}
```

```
unsigned int updcrc(int c, unsigned int crc)
{
  for(int i=7;i>=0;i--)
  {
    if(crc&0x8000)
    {
      crc<<=1;
      crc+=(((c<<=1)&0x100)!=0);
      crc^=0x1021;
    }
    else
    {
      crc<<=1;
      crc+=(((c<<=1)&0x100)!=0);
    }
  }
  return crc;
}
```