

Welcome to C programming for devices.

Arduino 2.

Topics for today:

- * **Use of a struct for DCC messages.**
- * **Protocols.**
- * **Timer constants.**
- * **Timer setup for interrupt.**
- * **The interrupt routine.**

Structures:

- * How to create a struct, insert data, read the data.

A struct:

```
* struct Message msg[MAXMSG] =  
* • {  
* • { { 0xFF, 0, 0xFF, 0, 0, 0, 0}, 3}, // idle msg  
* • { { 0, 0, 0, 0, 0, 0, 0}, 3}, // locoMsg short  
* • { { 0, 0, 0, 0, 0, 0, 0}, 4} // locoMsg long  
* •  
* • };
```

Protocols:

- * What protocols do you know ?
- * • English, German , Danish
- * • Ethernet, IEEE 802.11bgn, Token Ring, Bluetooth.
- * • RS232, RS422, RS485
- * • DCC

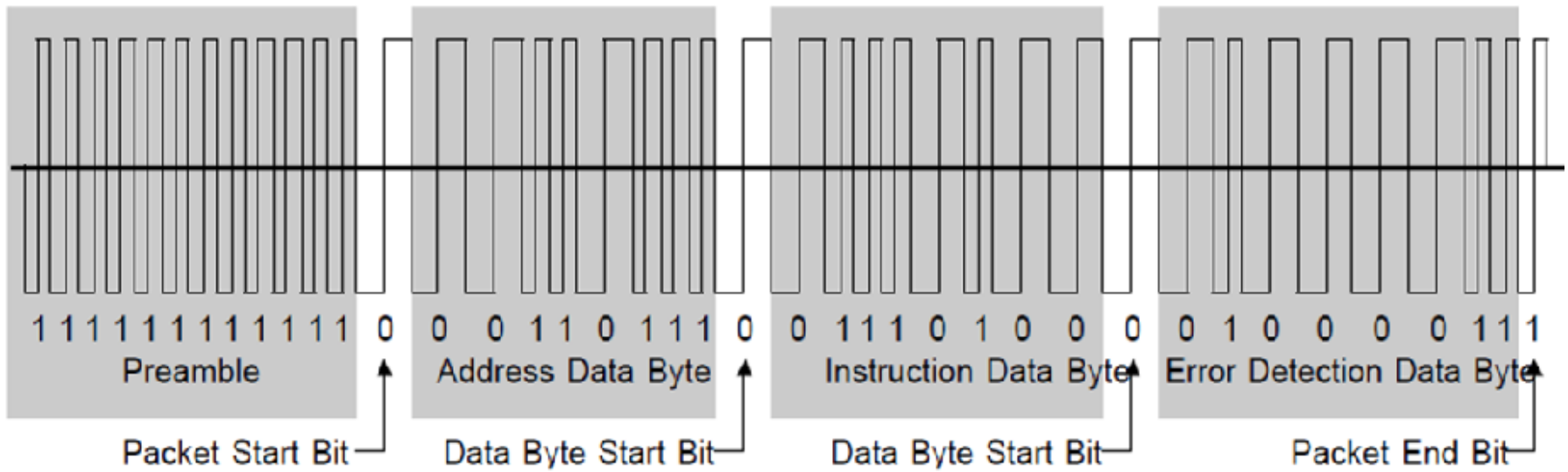
RS232:

- * What is important to RS232
- * • Bits, stopbits, parity, speed. Frame start, frame size.
- * • DCC
- * • Bits, frame start, separator bits, stop bit, pulswidths
58/116 microseconds
- * • Communitation with locomotive address, data.

RS232:



DCC:



Timer constants:

- * • //Timer frequency is 2MHz for (/8 prescale from 16MHz)
- * • #define TIMER_SHORT 0x8D // 58usec pulse length
- * • #define TIMER_LONG 0x1B // 116usec pulse length

Timer setup for interrupt:

```
* void SetupTimer2()  
* • {  
* • //Timer2 Settings: Timer Prescaler /8, mode 0  
* • //Timmer clock = 16MHz/8 = 2MHz oder 0,5usec  
* • TCCR2A = 0;  
* • TCCR2B = 0<<CS22 | 1<<CS21 | 0<<CS20;  
* • //Timer2 Overflow Interrupt Enable  
* • TIMSK2 = 1<<TOIE2;  
* • //load the timer for its first cycle  
* • TCNT2=TIMER_SHORT;  
* • }
```

The ISR routine

- * ISR(TIMER2_OVF_vect)
- * • {
- * • //Here is the code to be executed by the //interrupt routine
- * • // send a frame
- * • }

Assignment A6:

- * Connect a button to an input port.
- * • Press the button to turn on the timer.
- * • Press again to stop the timer.
- * • Who in your group are closest to one minute ?,
- * • Who is the best in the class ?.

Assignment A7:

- * Connect the keypad to the arduino, write a number, display it with `Serial.print()`.
- * • Now let the arduino find a random number 0 – 255.
- * • With the keypad guess the secret number, when you succeed, let the arduino show
- * • How many trials you had.
- * •

Assignment A8:

- * Create a program where an output pin gives pulses at 58 and 116 Microseconds.
- * • Use a delay function to do the work

Assignment A9:

- * By help of the timerinterrupt generate a puls with the pulswide of 58 microseconds.
- * • After that change the timer to 116 microseconds

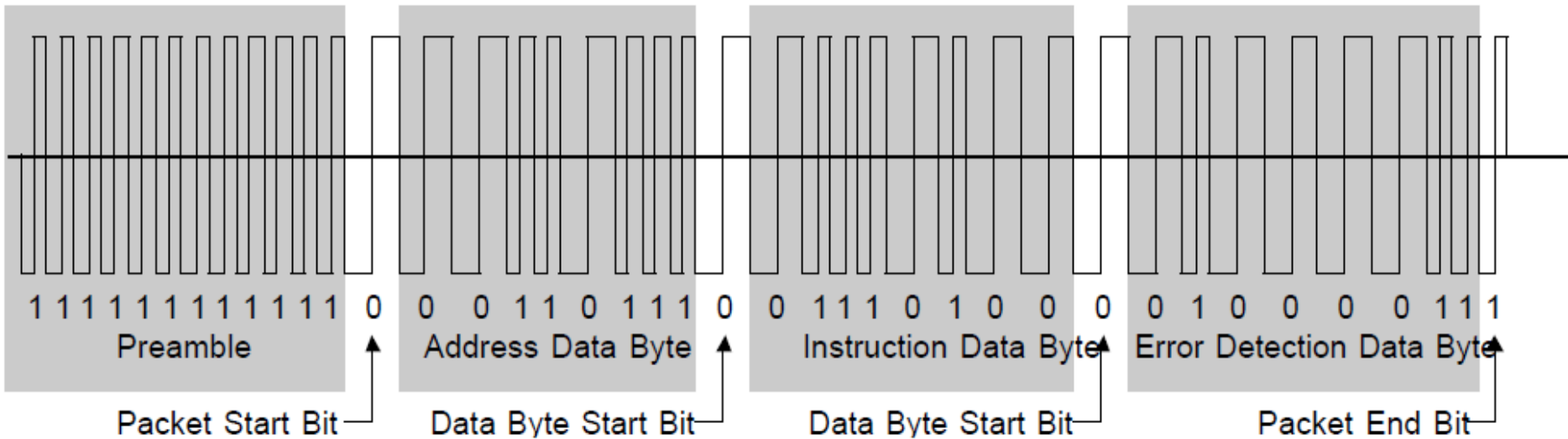
Assignment A9:

- * An address starts always with a zero as MSB.
- * An example could be B00100100 (36)
- * A command could be 0110 0100
- * Errorbyte = (adresse byte ^ kommando byte).
- * Each byte is separated by a Zero.
- * 1 is pulses of 58 microseconds (per halfcycles).
- * 0 is pulses of 116 microseconds (per halfcycle).

The command byte:

- * B01100110 Forward speed 6
- * B01101010 Forward speed 10
- * B0110 0000 Stop
- * B00100000 Stop
- * B00100110 Backwards speed 6
- * B00101010 Backwards speed 10

The DCC frame:



Try to make a frame :

- * By help of interrupt try to make a frame for our train.
- * Check the frame with a signal analyzer.
- * If it looks ok, then connect to the booster, and try to control the train.