FDUP (Fermcontroller Display Update Protocol) ver. 1.0

The FDUP protocol is a communication protocol used to relay display data to microcontroller units that control the 7-segment display clusters on the Fermcontroller HMI.

The display clusters of the original Fermcontroller system are 4-digit clusters, so the data frame size in this original protocol specification (version 1.0) is 11 octets. The protocol is, however, easy to expand to support bigger display clusters if needed.

The original Fermcontroller HMI implementation uses AVR Attiny861 MCU's to control the display clusters and communicates with them over SPI. The Attiny's implement SPI communications with the USI (Universal Serial Interface), which doesn't have hardware support the the Slave Select signal. All of the GPIO pins were also tied up for controlling the display segments, and thus Slave Select could not be implemented in software, either. The missing Slave Select functionality had to be replaced with device addressing within the FDUP protocol. The slave devices will only update their display contents if the Slave address byte matches their assigned address. Otherwise they will ignore the data frame.

FDUP data frame description

Position	Description	Note	
Octet 1	Frame start byte	Always 0xBA	Frame Header
Octet 2	Slave address	0x0 … 0xFF	
Octet 3	Display 1 character	Refer to Character Translation Table	Display contents
Octet 4	Display 1 Decimal Point Enable	0x0 for no decimal point, 0x1 for decimal point	
Octet 5	Display 2 character	Refer to Character Translation Table	
Octet 6	Display 2 Decimal Point Enable	0x0 for no decimal point, 0x1 for decimal point	
Octet 7	Display 3 character	Refer to Character Translation Table	
Octet 8	Display 3 Decimal Point Enable	0x0 for no decimal point, 0x1 for decimal point	
Octet 9	Display 4 character	Refer to Character Translation Table	
Octet 10	Display 4 Decimal Point Enable	0x0 for no decimal point, 0x1 for decimal point	
Octet 11	Frame end byte	Always 0xBE	Frame end

FDUP Character Translation Table

Character to display	Protocol byte
Empty display	0x00
'0'	0x10
'1'	0x11
'2'	0x12
'3'	0x13
'4'	0x14
'5'	0x15
'6'	0x16
'7'	0x17
'8'	0x18
'9'	0x19
'A'	0x1A
'C'	0x1B
'E'	0x1C
'F'	0x1D
'H'	0x1E
'I'	0x1F
'J'	0x20
'L'	0x21
'O'	0x22
'P'	0x23
'S'	0x24
'U'	0x25
Minus sign	0x26