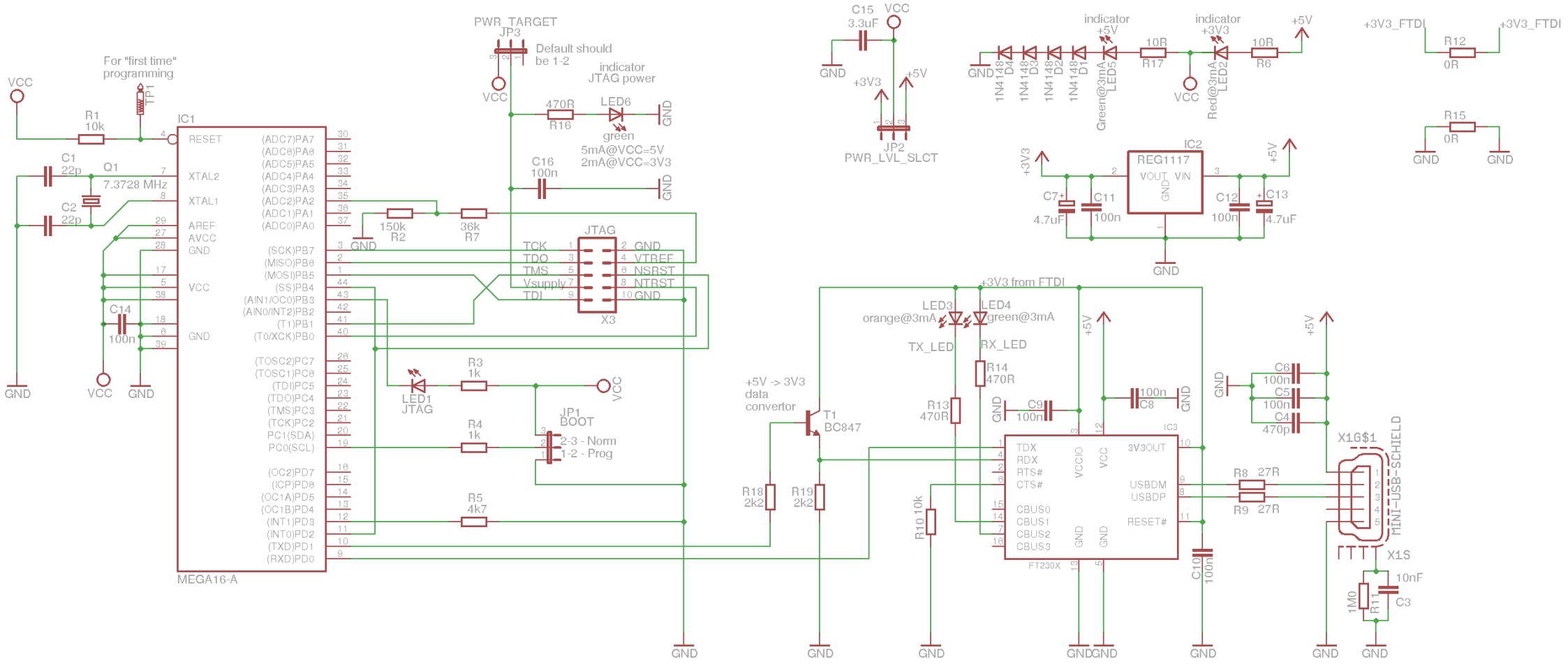
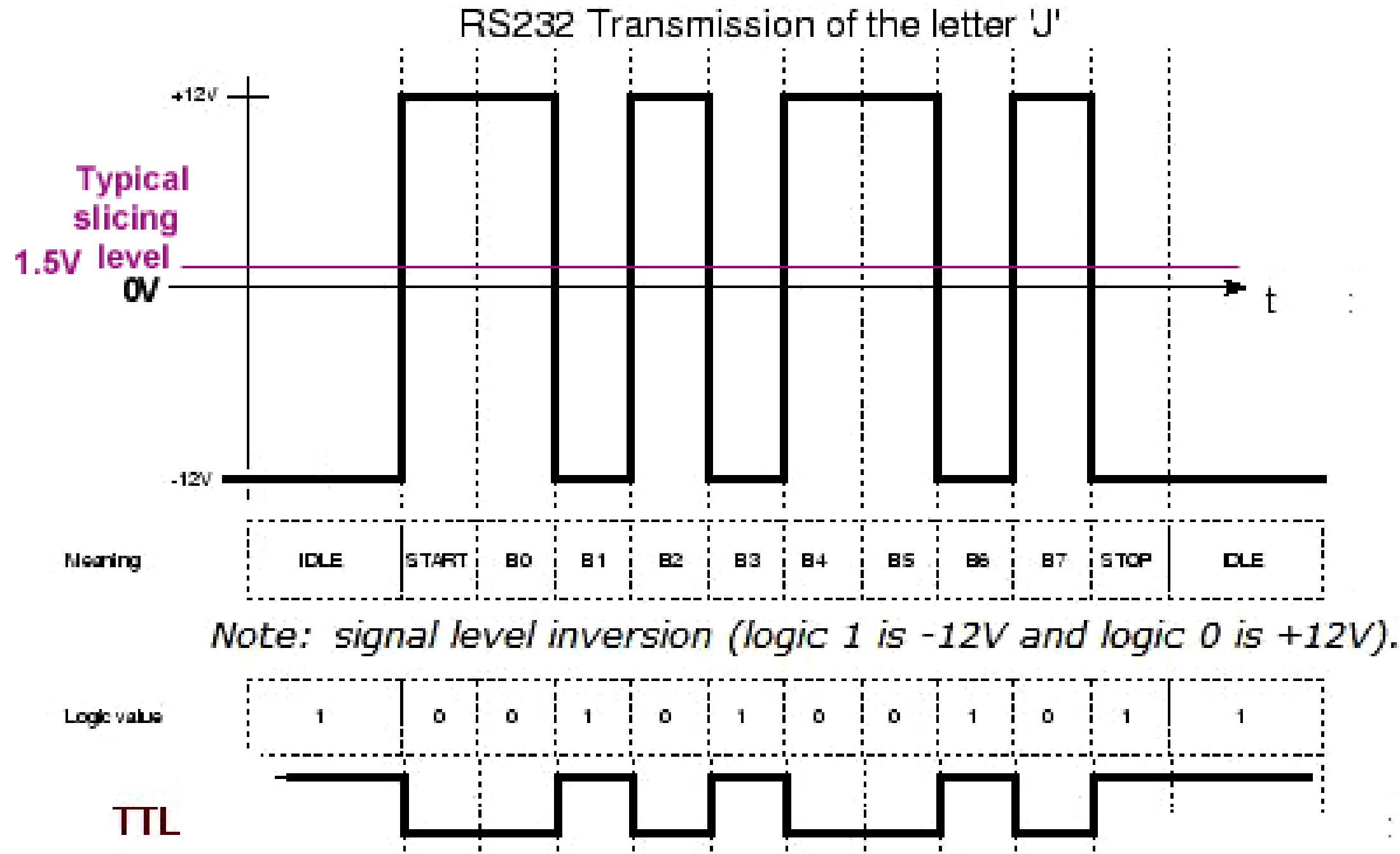
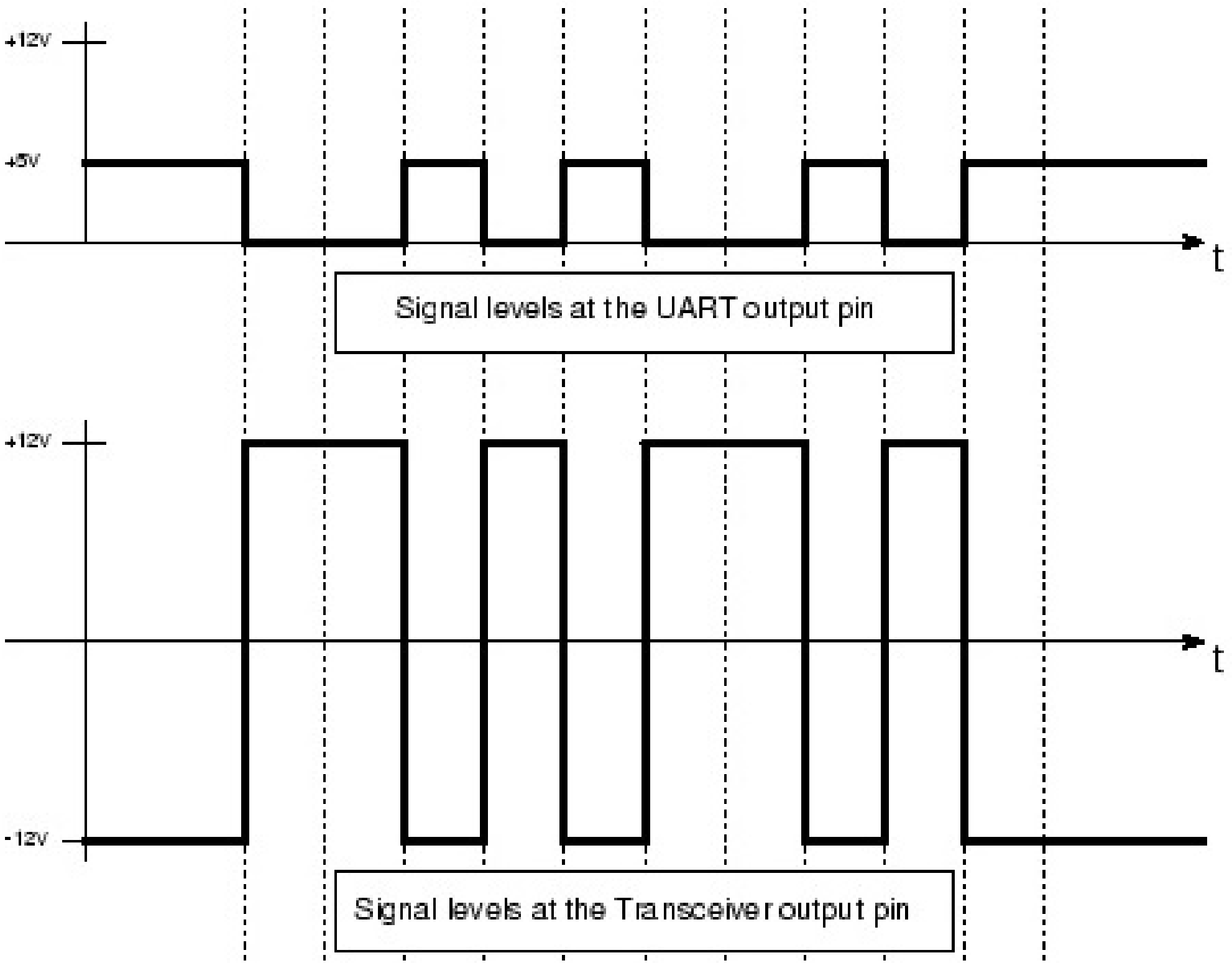

For "first time" programming
TP1
VCC
R1
10k
IC1
RESET
XTAL2
XTAL1
AREF
AVCC
GND
VCC
GND
C1
22p
Q1
7.3728 MHz
C2
22p
C14
100n
GND
VCC
GND
(ADC7)PA7
(ADC6)PA6
(ADC5)PA5
(ADC4)PA4
(ADC3)PA3
(ADC2)PA2
(ADC1)PA1
(ADC0)PA0
(SCK)PB7
(MISO)PB6
(MOSI)PB5
(SS)PB4
(AIN1/OC0)PB3
(AIN0/INT2)PB2
(T1)PB1
(T0/XCK)PB0
(TOSC2)PC7
(TOSC1)PC6
(TDI)PC5
(TDO)PC4
(TMS)PC3
(TCK)PC2
PC1(SDA)
PC0(SCL)
(OC2)PD7
(ICP)PD6
(OC1A)PD5
(OC1B)PD4
(INT1)PD3
(INT0)PD2
(TXD)PD1
(RXD)PD0
MEGA16-A
PWR_TARGET
JP3
Default should be 1-2
VCC
470R
R16
LED6
green
indicator JTAG power
5mA@VCC=5V
2mA@VCC=3V3
GND
C16
100n
GND
150k
R2
36k
R7
GND
JTAG
TCK
TDO
TMS
Vsupply
TDI
GND
VTREF
NSRST
NTRST
GND
X3
LED1
JTAG
R3
1k
VCC
JP1
BOOT
2-3 - Norm
1-2 - Prog
R4
1k
R5
4k7
GND
C15
3.3uF
VCC
GND
+3V3
+5V
JP2
PWR_LVL_SLCT
indicator +5V
indicator +3V3
+5V
GND
1N4148 D4
1N4148 D3
1N4148 D2
1N4148 D1
Green@3mA LED5
10R
R17
Red@3mA LED2
10R
R6
VCC
+3V3_FTDI
+3V3_FTDI
R12
0R
R15
0R
GND
GND
+3V3
IC2
REG1117
VOUT
VIN
GND
+5V
C7
4.7uF
C11
100n
C12
100n
C13
4.7uF
GND
+3V3 from FTDI
LED3
orange@3mA
LED4
green@3mA
TX_LED
RX_LED
R14
470R
R13
470R
+5V -> 3V3
data
convertor
T1
BC847
R18
2k2
R19
2k2
R10
10k
GND
C9
100n
+5V
100n
C8
GND
IC3
TDX
RDX
RTS#
CTS#
CBUS0
CBUS1
CBUS2
CBUS3
VCCIO
VCC
GND
GND
3V3OUT
USBDM
USBDP
RESET#
FT230X
C10
100n
R8
27R
R9
27R
GND
C6
100n
C5
100n
C4
470p
+5V
X1G$1
MINI-USB-SCHIELD
X1S
1M0
R11
10nF
C3
GND
GND
GND
GND
GND
GND
GND
GND

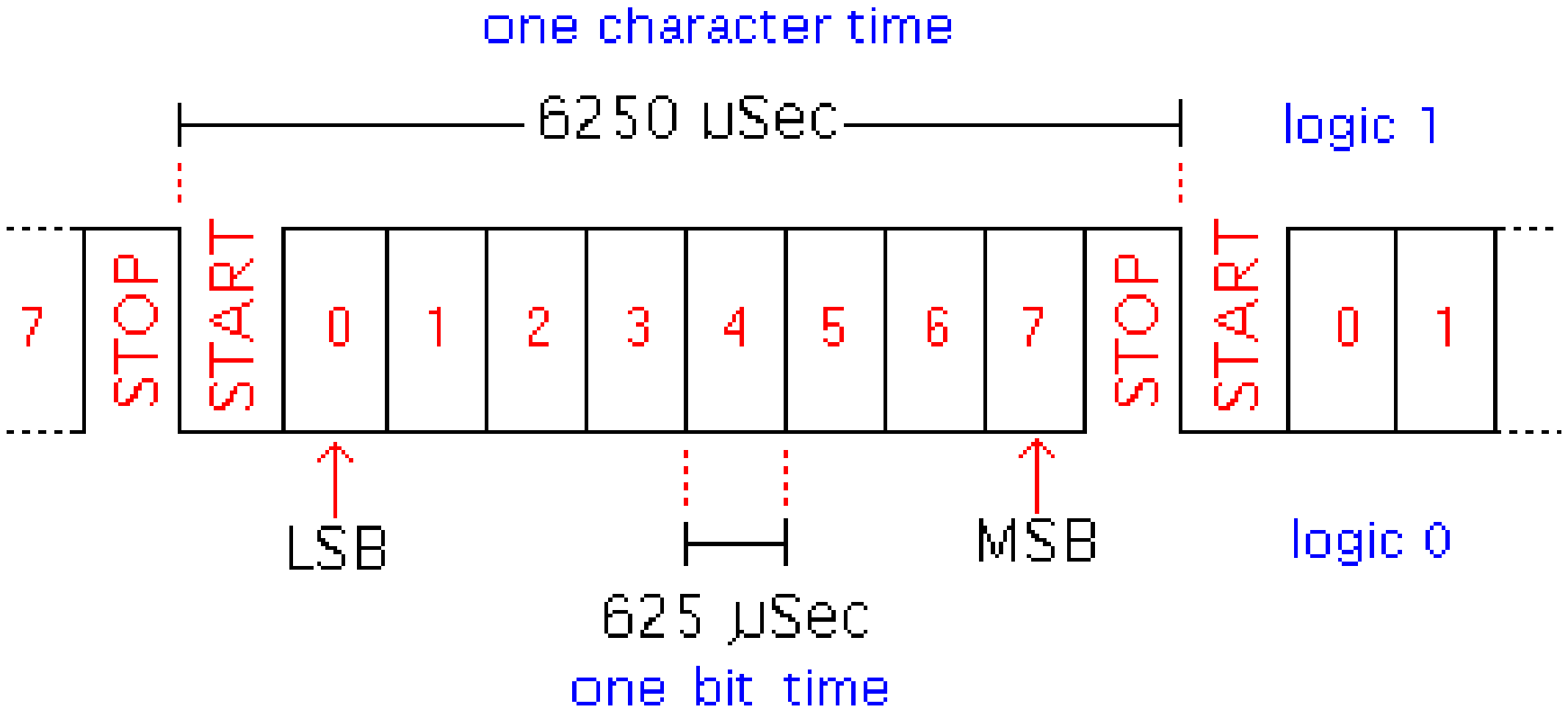

RS232 Transmission of the letter 'J'
+12V
Typical slicing
1.5V level
0V
-12V
t
Meaning
IDLE
START
B0
B1
B2
B3
B4
B5
B6
B7
STOP
IDLE
Note: signal level inversion (logic 1 is -12V and logic 0 is +12V).
Logic value
1
0
0
1
0
1
0
0
1
0
1
1
TTL

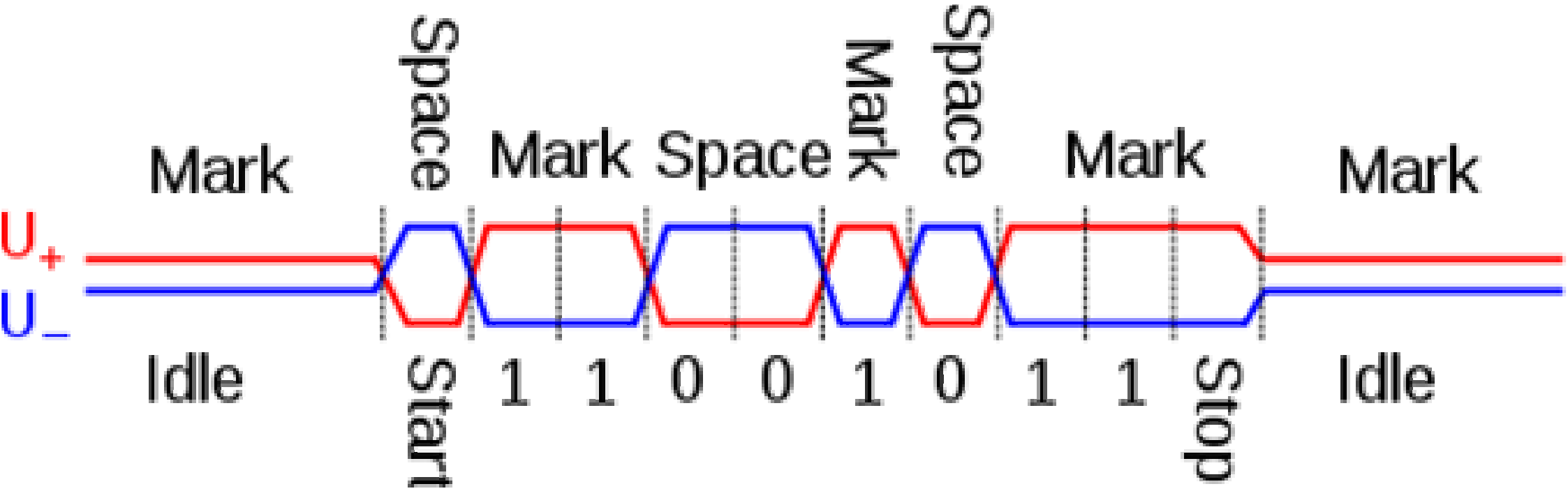

+12V
+5V
t
Signal levels at the UART output pin
+12V
-12V
t
Signal levels at the Transceiver output pin

HOW UART COMMUNICATES?

- ***START BIT***
- The UART data transmission line is normally held at a high voltage level when it's not transmitting data.
- To start the transfer of data, the transmitting UART pulls the transmission line from high to low for one clock cycle.
- When the receiving UART detects the high to low voltage transition, it begins reading the bits in the data frame at the frequency of the baud rate.

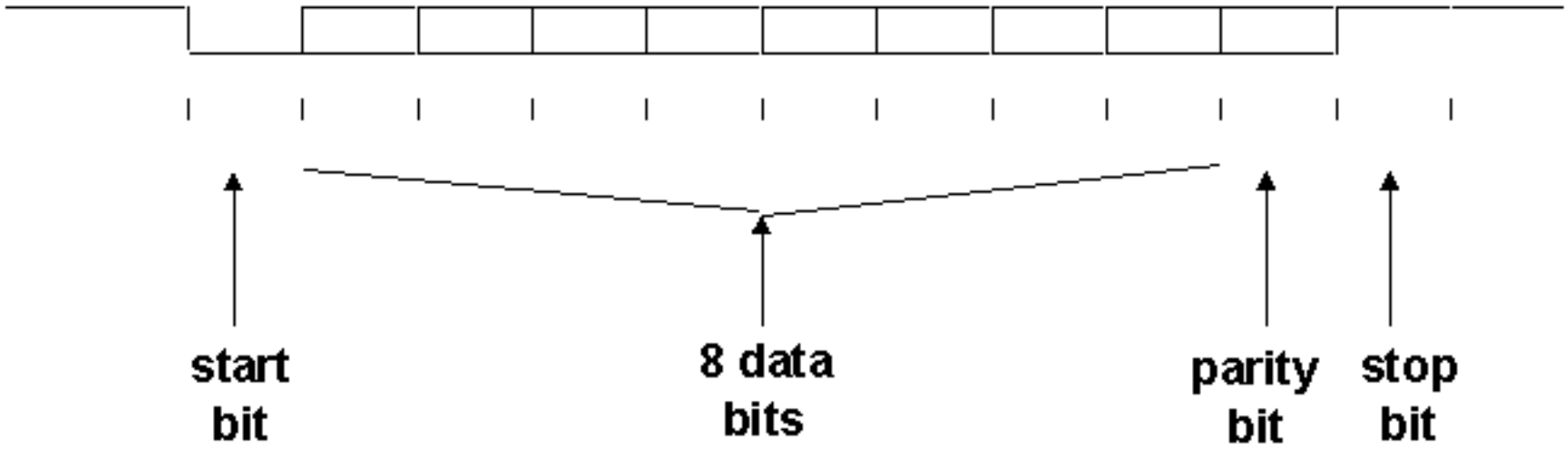
1600 baud Serial

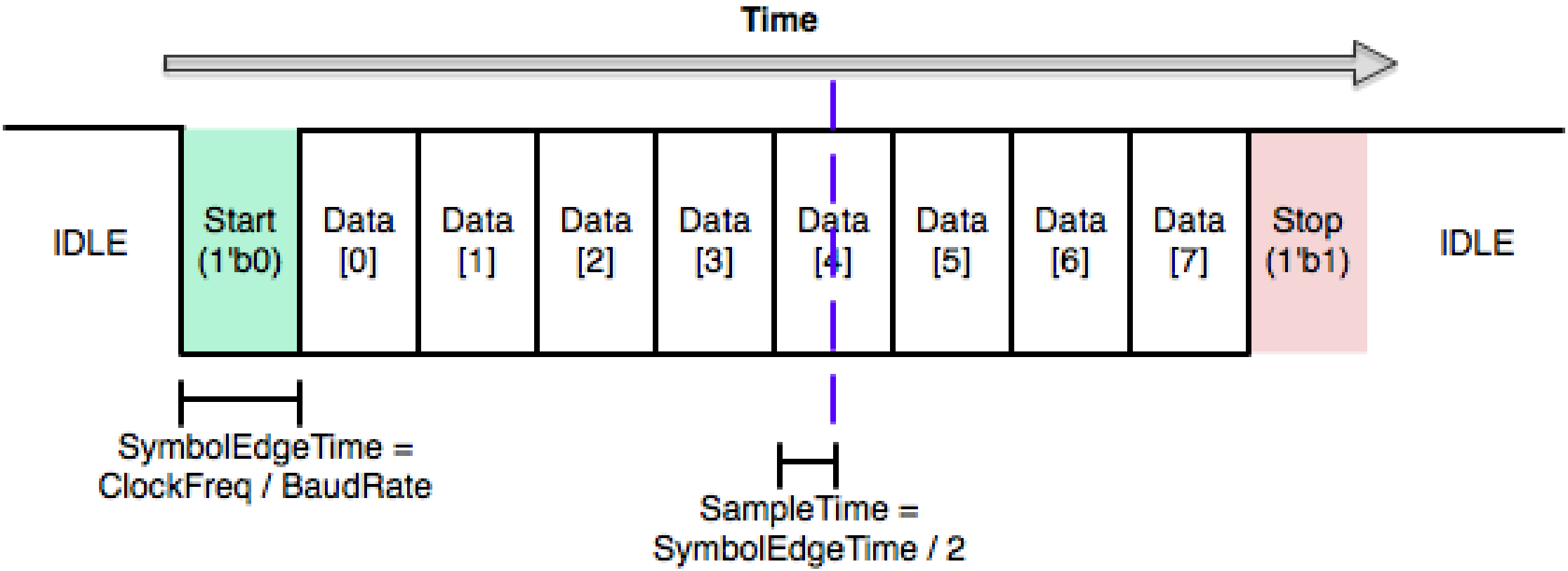


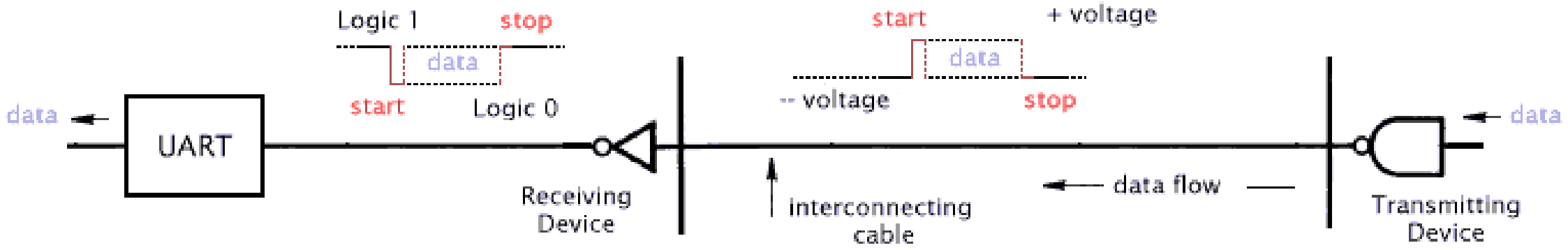

Mark
Space
Mark
Space
Mark
Space
Mark
Mark
U+
U_
Idle
Start
1
1
0
0
1
0
1
1
Stop
Idle

Asynchronous RS-232: data format

- Variations: Parity bit; 1, 1.5, or 2 stop bits
- Must oversample data (16× is typical) to find bit boundaries
  - Start bit is crucial
  - UART does the synchronization




Time
IDLE
Start
(1'b0)
Data
[0]
Data
[1]
Data
[2]
Data
[3]
Data
[4]
Data
[5]
Data
[6]
Data
[7]
Stop
(1'b1)
IDLE
SymbolEdgeTime =
ClockFreq / BaudRate
SampleTime =
SymbolEdgeTime / 2


Logic 1
stop
data
start
Logic 0
start
+ voltage
data
-- voltage
stop
data
UART
Receiving
Device
interconnecting
cable
data flow
Transmitting
Device
data

160 baud ALDL vs 1600 baud Serial