EMBEDDED SYSTEMS 2017/2018: C4uC [6/9 credits course]

Max total: 12 points

Available time: 90 minutes

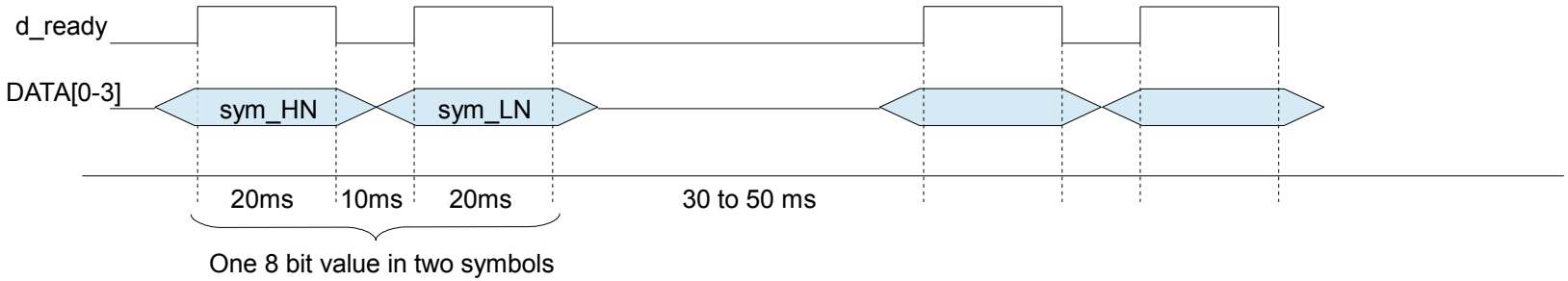
Name:_____

C4uC (with reference to *uC Atmel 328p*)

N.B.
The main design choices shall be described and justified.

C4µC - Examination

- A microcontroller system is required to process some data coming on a parallel bus
  - The bus has 5 lines (4 bit + d_ready), but transfers 8 bit values (uint8_t) in two symbols, according to the time diagram below



  - The two symbols carry high and low nibble of the 8 bit value

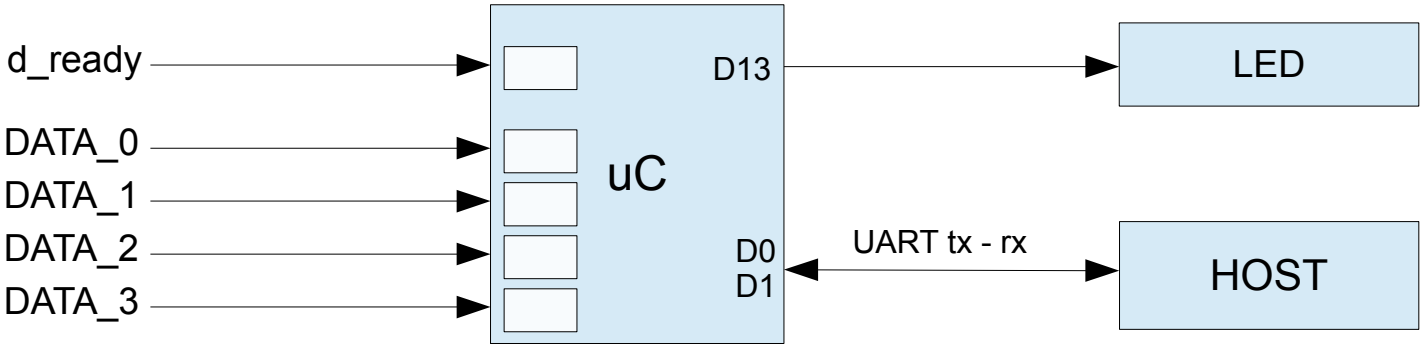
Bus bit	sym_HN	sym_LN
DATA_0	b4	b0
DATA_1	b5	b1
DATA_2	b6	b2
DATA_3	b7	b3

sym_HN				sym_LN			
b7	b6	b5	b4	b3	b2	b1	b0
uint8_t value (0 - 255)							

  - Note that the interval between each couple of symbols is variable from 30 to 50 ms.

C4µC - Examination

- Partial connection schematic:



- Exercise tasks...:

1) Define, completing the schematic, appropriate inputs for the bus signals, considering the reference µC 328p and that the data lines on the bus must be sampled at the same time [up to 2pts]

2) Develop a C program that receives sequences of 10 values, with:

   - The signalization of sequence complete on the LED (turn ON 1s, then OFF) [up to 2pts]
   - The signalization of error if time between values is outside interval [30,50] ms (the error code is 5 blinks at 5 Hz); this event closes the sequence!!! [up to 2pts]

C4µC - Examination

- ... and then:

3) Design a function that calculates the first 100 numbers of the Fibonacci sequence, i.e.: [up to 2pts]
   - 1, 2, 3, 5, 8, ... 144, 233, **121**, **98**, ...
     but modulo 256 (truncates them at precision of an uint8_t)

4) Complete or redesign the two functions given below to store and check such numbers [up to 2pts]

```
uint8_t LUT[32] = {0};   // LookUpTable where the Fibonacci numbers are stored

void store_number(uint8_t num){  // store function
  uint8_t idx = num >> 3;
  uint8_t shift = num & 0x07;
  LUT[idx] = ...
}

bool check_number(uint8_t num){   // check function
  uint8_t idx = num >> 3;
  uint8_t shift = num & 0x07;
  return ...
}
```

5) Modify the C program (from task 2) so that:
   - If a received value is a Fibonacci number, it is sent over the UART to the host as a uint8_t (not a string representation of the number) [up to 2pts]