Al centro del sistema di sviluppo vi è la sezione controllata via UART e che fa capo al controllore. Questo sistema permette di controllare le funzionalità della scheda inviando comandi via UART alla scheda stessa. Questo permette di non dover ricaricare il design sull’FPGA per modificare le memorie o la frequenza del clock.

Tutte le unità presenti le ho progettate e implementate perché non avessero necessità di interfacce ulteriori e con lo scopo di massimizzare la velocità. La possibilità del debug e del controllo sono garantite se le operazioni che compie il controllore e le altre macchine rientrino all’interno di mezzo ciclo di CLK così che lo Z80X non riesca a notare le modifiche. L’unità che fa interfaccia per il protocollo UART invece l’ho usata come un IP da “Simple UART for FPGA” di Jakub Cabal [https://github.com/jakubcabal/uart-for-fpga].

L’interfaccia UART non permette molti controlli sulla gestione della comunicazione ma è ideale per quello che deve fare all’interno di questo sistema. Permette di selezionare la velocità di trasmissione, detta baudrate, con il generic Baud\_Rate e la presenza e la tipologia del bit di parità, detto parity bit. Per questo design ho impostato la prima al massimo valore accettato dal driver USB cioè 115200 baud/s e nessun parity bit per cui la trasmissione di un carattere dura circa 87 us.

Il port dell’interfaccia mostra due port, uno per l’invio e l’altro per la ricezione.  
Per l’invio, è presente il bus per i dati, DIN, assieme ai segnali DIN\_VLD e DIN\_RDY. Con il primo si segnala all’interfaccia che il valore presente su DIN è stabile e può essere inviato, mentre con DIN\_RDY l’interfaccia comunica di essere pronta per il prossimo invio.  
Per l’uscita, sono presenti alcuni segnali che indicano errori nei valori ricevuti però di interesse sono i segnali DOUT e DOUT\_VLD. DOUT è il bus che contiene il valore ricevuto e solamente quando avviene un impulso su DOUT\_VLD si può considerare il suo valore corretto. Per cui si può usare DOUT\_VLD come impulso di scrittura sulla FIFO\_RX.