



Der Ez.Slo-BUS

In den Zeiten während viele BUS-Systeme (IEEE 488 Bus, CAN, I²C, ProfiBUS, etc.) entwickelt wurden, waren Speicherplatz teuer und Reaktionszeiten kritisch. Der heutige Ez.Slo-Bus ist einfach, benötigt aber etwas Speicherplatz und wurde entwickelt für langsame Systeme, bei denen nur geringe Datenmengen übertragen werden müssen.

Die Nachrichten auf dem Bus, die aus bis zu 19 Bytes bestehen, werden vom Host zum Client und vom Client zum Host gesendet. Der Host bestimmt die Richtung der Übertragung; d.h ein Client darf eine Übertragung nicht unterbrechen.

Die seriell übertragenen Bytes werden von einem UART generiert und bestehen aus einem START Bit, acht Informationsbits, und einem STOP Bit. Ein Parity-Bit fehlt. Die Übertragungsrate ist auf 19,2 kBaud festgelegt. RS 485 Bus-Transceiver treiben die 120 Ω differentielle Leitung, deren Länge auf 15 Meter beschränkt ist.

Das erste Byte der Nachricht enthält die Adresse des ausgewählten Clients; das zweite Byte der Nachricht enthält einen Zeichen, auf den der Client reagiert. Nach den ersten zwei Byte folgen von null bis zu maximal 16 weitere Byte, denen ein Carriage Return (0x0D) als „Delimitator“ folgt. Nur darstellbare ASCII Zeichen sind erlaubt.

Nachrichtenübertragungen zwischen den Clients sind momentan nicht vorgesehen (Mai 2012).

Der Host ist im Normalfall in Sendebereitschaft; die Clients in Empfangsbereitschaft.

Das erste Byte einer Nachricht enthält die Adresse des Clients, der auf die Nachricht reagieren soll. Das Most Significant Bit (MSB) des ersten Bytes, der Adresse, ist immer Hi; die niedrigste Adresse ist 0x21 (ASCII >|<), d.h. 0xA1 und die höchste ist 0x7F, d.h. 0xFF. So können also bis zu 95 verschiedene Clients adressiert werden.

Das zweite Byte – sein Wert geht von 0x40 bis 0x7F – gibt an, wie der angesprochene Client auf die folgenden Bytes reagieren soll. Werden nur zwei Byte {plus Carriage Return (0x0D) als „Delimitator“} vom Host gesendet, ist der angesprochene Client aufgefordert sofort und in dem vorgesehenem Zeitfenster maximal 16 Bytes plus Carriage Return (0x0D) als „Delimitator“ an den Host zu senden.

Nachdem alle Clients empfangsbereit sind, untersuchen alle Clients ob das MSB des ersten Bytes einer Nachricht Hi ist. Ist das der Fall, ist das erste Byte ein Adresse; falls nicht, wird die Nachricht ignoriert. Stimmt die gesendete Adresse mit der eines Klienten überein, überprüft dieser ob nur zwei Byte empfangen wurden. Clients bei denen die Adressen nicht übereinstimmen ignorieren die Nachricht.

Werden nur zwei Bytes empfangen, reagiert der Client auf das zweite Byte und sendet die vereinbarten Bytes zum Host. Das zweite Byte ist größer oder gleich 0x60 (ASCII >`<) und kleiner oder gleich 0x7D (ASCII >}<). Falls der Client nicht innerhalb eines bestimmten Zeitrahmens die Antwort

sendet, registriert der Host ein Time-out und schaltet von der Empfangsbereitschaft in die Sendebereitschaft.

Ist das zweite Byte ein:

- 0x61 >a<, dann sendet der Client als Antwort seine Artikel-Nummer.
- 0x69 >i<, dann sendet der Client als Antwort seine Identifikation.
- 0x6D >m<, dann sendet der Client als Antwort den Herstellernamen.
- 0x73 >s<, dann sendet der Client als Antwort seine Serien-Nummer.
- 0x76 >v<, dann sendet der Client als Antwort seine Software-Version.
- 0x7E >~<, dann sendet der Client als Antwort >Client ? Active< wo >?< ein Platzhalter für die Adresse des Clients (mit MSB = Lo) ist.

Werden in der Nachricht vom Host mehr als zwei Bytes gesendet, reagiert der angesprochene Client auf das zweite Byte und verarbeitet die empfangenen Bytes entsprechend. Das zweite Byte ist größer oder gleich 0x40 (ASCII >@<) und kleiner oder gleich 0x5F (ASCII >_<).

Ist das zweite Byte entweder 0x41 (ASCII >A<) oder 0x53 (ASCII >S<) reagiert der angesprochene Client wie folgt:

Commands must be preceded
by the clients address

| | | | | | | | |
|------|------|------|------|------|------|------|------|
| ! | " | # | \$ | % | = | > | ? |
| 0x21 | 0x22 | 0x23 | 0x24 | 0x25 | 0x3D | 0x3E | 0x3F |

| Command (ASCII) | Action |
|-----------------|---|
| "Axy" | Clients Address (default 0x30 ASCII >0<) can be programmed. where >x< and >y<, having equal values greater than or equal to 0x21 (ASCII >!<) and less than or equal to 0x3F (ASCII >?<) |
| "Sxxxxxxxx" | Module Serial Number is programmed only once. where xxxxxxxx is greater than ASCII >00000000< and less than or equal to ASCII >99999999< |

Das Flußdiagramm:

