

Leider kann ich die pdf nicht bearbeiten, daher hier meine Änderungen.

Ort	Original	Änderung
Seite 4 Beschreibung Receive/Receive_Enable	<p>Der erste Schritt (das Empfangen) lässt sich wiederum in drei Schritte unterteilen. Es beginnt mit der Einstellung des USART auf Empfangen. Dazu müssen, im UCR, die Bits 7, 6 und 3 auf 0 gesetzt werden und das Bit 4 auf 1. Heißt: Übertragung aus, Interrupt für die Benachrichtigung empfangener und gesendeter Daten aus und Empfangen ein. Danach geht er in StartSequence über.</p>	<p>Der erste Schritt (das Empfangen) lässt sich wiederum in drei Schritte unterteilen. Es beginnt mit der Einstellung des USART auf Empfangen <b>(RECEIVE_ENABLE)</b>.</p> <p><b>Dazu werden die Bits TXEN, TEXCIE und RXCIE gelöscht und RXEN gesetzt. Somit ist das Senden und der entsprechende Interrupt, sowie der Interrupt bei Empfang eines Zeichens abgeschaltet.</b></p> <p>Danach geht er in StartSequence über.</p>
Seite 5 Beschreibung Startsequence	<p>Die StartSequence beginnt mit dem sichern des Registers R16 und des Z-Pointers. Nachdem der Zähler auf Null gesetzt wurde und die Adressen zugewiesen wurden beginnt StartSequence_1. Hier wird zuerst überprüft ob das Bit 7 des UCSRA gesetzt ist. Ist es das nicht und RXC enthält eine 0 ist noch kein Zeichen empfangen worden. In diesem Fall springt der PC wieder zurück zu StartSequence_1 und hängt somit in einer Warteschleife bis ein Zeichen empfangen wurde.</p>	<p>Die StartSequence beginnt mit dem sichern des Registers R16 und des Z-Pointers. Nachdem der Zähler auf Null gesetzt wurde und die Adressen zugewiesen wurden beginnt StartSequence_1.</p> <p><b>Hier wird solange gewartet bis ein Zeichen empfangen wurde. Dies geschieht durch Abfrage des Bits RXC (Receive Complete) . Ist dieses gesetzt, so wurde ein Zeichen empfangen. Dieses wird nun aus dem UDR ausgelesen und an der Speicheradresse des Z-Pointers abgelegt. Anschließend wird der Z-Pointer um eins erhöht. Des Weiteren wird der Zähler „count“ ebenfalls um 1 erhöht. Dieser Zähler dient als Zähler der empfangenen Zeichen.</b></p>

	<p>Ist ein Zeichen empfangen worden wird der Inhalt des Empfangsregisters gespeichert und die Speicheradresse für das nächste Zeichen weiter geschoben. Nun wird überprüft ob schon 3 Zeichen gesendet wurden. Dies dient dem Zweck der Überprüfung ob es sich bei den drei Zeichen um die Startsequenz handelt. (checkStartSequence wird auf der nächsten Seite beschrieben) Handelt es sich bei den geprüften drei Zeichen um die Startsequenz werden die Register wieder hergestellt und es geht mit RX_INFO weiter. Sollte es sich nicht um die Startsequenz gehandelt haben geht es mit der Funktion shiftSequence weiter.</p>	<p>Der nächste Schritt ist die Prüfung, ob bereits mehr als drei Zeichen empfangen wurden. Ist dies nicht der Fall springt das Unterprogramm zu STARTSEQUENCE_1 zurück und wartet auf den Empfang des nächsten Zeichens. Sollten jedoch mehr als drei Zeichen empfangen worden sein, so wird nun geprüft, ob es sich um die Startsequenz handelt. Dazu wird das Unterprogramm „checkSequence“ aufgerufen. Dieses wird im weiteren Verlauf der Dokumentation noch näher beschrieben.</p> <p>Ist die Startsequenz nicht empfangen wurden, so wird in das Unterprogramm shiftSequence gesprungen, welches ebenfalls im weiteren Verlauf noch näher beschrieben wird. Anschließend wird der Z-Pointer dekrementiert. Die Ursache dafür ist im Unterprogramm „shiftSequence“ zu suchen. Weiter geht es nun mit dem Rücksprung zu STARTSEQUENCE_1 um auf den Empfang des nächsten Zeichens zu warten.</p> <p>Wurde jedoch die Startsequenz empfangen, so wird das Unterprogramm „StartSequence“ beenden. Dazu werden lediglich die am Anfang gesicherten Register wiederhergestellt. Nach dem Rücksprung in das aufrufende Unterprogramm „RECEIVE“ springt dieses in das Unterprogramm „RX_INFO“. Dieses hat die Aufgabe die eigentliche Nachricht zu empfangen und zu speichern.</p>
--	---	---

Seite 7 untere Hälfte	Nachteil liegt darin, wenn eine Zeichenfolge empfangen wird, welche die Startsequence beinhaltet, aber <b>noicht</b> für den ASURO gedacht ist. Abgefangen wird dies durch das Timeout in RX_INFO, solange die Zeichenkette nicht mehr als 250 Zeichen beinhaltet.	Nachteil liegt darin, wenn eine Zeichenfolge empfangen wird, welche die Startsequence beinhaltet, aber <b>nicht</b> für den ASURO gedacht ist. Abgefangen wird dies durch das Timeout in RX_INFO, solange die Zeichenkette nicht mehr als 250 Zeichen beinhaltet.
Seite 8 letzter Absatz	In TIMEOUT_ON wird nach der Registersicherung Prescaler auf etwa 1s eingestellt. Danach werden die Register wieder hergestellt und es erfolgt der Rücksprung in die aufrufende Funktion.	In TIMEOUT_ON wird nach der Registersicherung <b>der</b> Prescaler auf etwa 1s eingestellt. Danach werden die Register wieder hergestellt und es erfolgt der Rücksprung in die aufrufende Funktion.
Seite 11 letzter Absatz	Nun wird ein Warten aktiviert, welches die weiter oben oder rechts beschriebene Funktion erfüllt. Danach werden die gesicherten Register wieder hergestellt.	Nun wird ein Warten aktiviert, <b>welches das zeitlich das komplette überfahren der LEDs sichert</b> . Danach werden die gesicherten Register wieder hergestellt.
Seite 12 erster Absatz	Nun kann der Asuro mit der Linienverfolgung starten. Nach dem sichern der verwendeten Register wird die Linienverfolgung initialisiert. Hierzu werden die Tabellen für die Regler in den dafür vorgesehen Bereichen abgespeichert. Die Größe des Speicherplatzes wurde zu Beginn des Gesamtprogramms mit 8 Byte je Regler definiert. Zudem werden die Messwerte des linken und rechten Sensors (mr und ml) 1 gesetzt. Sollten diese Messwerte 0 annehmen ist das Ziel erreicht. Des weiteren wird die Front LED	Nun kann der Asuro mit der Linienverfolgung starten. Nach dem sichern der verwendeten Register wird die Linienverfolgung initialisiert. Hierzu werden die Tabellen für die Regler in den dafür vorgesehen Bereichen abgespeichert. Die Größe des Speicherplatzes wurde zu Beginn des Gesamtprogramms mit 8 Byte je Regler definiert. Zudem werden die Messwerte des linken und rechten Sensors (mr und ml) 1 gesetzt. Sollten diese Messwerte 0 annehmen ist das Ziel erreicht. Des weiteren wird die Front LED

	<p>angeschalten. Danach wird der Timer0 auf die Wartezeit (definierter Wert) eingestellt und der Interrupt des Timer0 freigegeben. Nach dem wiederherstellen der gesicherten Register ist die Initialisierung des Linienfolgers beendet.</p>	<p>angeschalten. Danach wird der Timer0 auf die Wartezeit (definierter Wert) eingestellt und der Interrupt des Timer0 freigegeben. <b>Wird dieser ausgelöst, wird der Timer abgeschaltet und der Interrupt für den ADC eingeschaltet. Dieser Interrupt wird ausgelöst, wenn ein Messwert vorliegt. Dieser wird in der Interrupt Service Routine „ADCI“ der entsprechenden Auswertung für Messwert linker/rechter Sensor mit oder ohne LED zugeführt. Nach der Auswertung des Messergebnisses, wird der ADC-Interrupt abgeschaltet und der Timer-Interrupt zugeschaltet und der Timer neu initialisiert und diese Schleife beginnt von vorn.</b> Nach dem wiederherstellen der gesicherten Register ist die Initialisierung des Linienfolgers beendet.</p>