In Abbildung 4-11 links ist das Verfahren zum Empfang von Zeichen mit Polling dargestellt. Dabei wird zyklisch abgefragt, inwieweit Daten im UDR vorhanden sind und diese gegebenenfalls als Rückgabewert zurückgeliefert. Sollten keine Daten eintreffen, so würde das gesamte Programm in dieser Endlosschleife angehalten. Aus diesem Grund wurde in der konkreten Ausführung ein timeout eingefügt, welches die Funktion nach Ablauf einer einstellbaren Zeit verlässt.
Die in Abbildung 4-11 rechts dargestellte Interrupt-Routine hat den Vorteil, dass diese lediglich beim Empfang von Zeichen durch den Mikrocontroller aufgerufen wird. Das Hauptprogramm muss also nicht auf den Dateneingang warten und kann weiter abgearbeitet werden. Wird ein Zeichen empfangen, so wird dieses in einen globalen Buffer geschrieben. Ist eine bestimmte Anzahl von Zeichen eingegangen oder wurde ein bestimmtes Steuerzeichen empfangen, so wird dem Hauptprogramm mit Hilfe eines Flags signalisiert, dass die Zeichenkette komplett empfangen wurde und zur Verarbeitung bereit steht.
Für den speziellen Anwendungsfall der Kommunikation mit dem Mobilfunkgerät wird eine Mischform aus Polling und Interrupt-Routine eingesetzt. Wie in Abbildung 4-8 rechts ersichtlich, wird bei der Initialisierung die Standard-Ein- und Ausgabe mit Hilfe der Funktion fdevopen() umgeleitet. Das bedeutet, dass eine formatierte Ein- und Ausgabe mit den Funktionen wie printf() oder scanf() erfolgen kann. Als so genannte Low-Level-Funktionen werden zum Senden und Empfangen von Zeichen dabei die Funktionen uart_getc() und uart_receive() gebraucht. Die Interrupt-Routine zum Empfang von Zeichen wird hingegen genutzt, um den Eingang eines Anrufes abzufangen.
Das eingesetzte Mobilfunkgerät kann mit Hilfe von Hayes AT-Kommandos über die serielle Schnittstelle gesteuert werden. So ist es z. B. mit dem Kommando AT+CHUP möglich, einen ankommenden Anruf abzuweisen. In dieser Arbeit ist es nicht möglich, alle Kommandos ausführlich zu erklären. Es werden vielmehr lediglich die genutzten Kommandos kurz dargelegt. Für eine ausführlichere Beschreibung sei an dieser Stelle auf die Quelle [5] verwiesen.
Im nächsten Kapitel wird erläutert, wie die soeben beschriebenen Funktionen eingesetzt werden, um mit Hilfe der AT-Kommandos eine Kurznachricht zu versenden.

4.4 Versenden von Daten als Kurznachricht im PDU Modus (Protocol Discription Unit) [2], [5]

Zum Versenden von Kurznachrichten ist lediglich ein einziges AT-Kommando notwendig, „AT+CMGS“. Die Schwierigkeit beim Versenden stellt die Kodierung der Textnachricht in das PDU-Format dar.
Eine Kurznachricht mit dem Inhalt „Hallo Welt“ an die Nummer 01621234567 sieht z. B. folgendermaßen aus:

AT+CMGS=22<cr>
0001000B811026214365F700000AC8309BFD065DCB6C3A<sub>

Die Zeichen <cr> und <sub> sind dabei Steuerzeichen, welche das Mobiltelefon benötigt. Das Steuerzeichen <cr> stellt dabei ein new line (dezimal 10) mit anschließendem carriage return (dezimal 13) dar und <sub> ein so genanntes substitute (dezimal 26).
Im Folgenden soll nun einzeln erläutert werden, inwieweit die Nachricht „Hallo Welt“ in das PDU-Format kodiert werden kann.
Nach dem AT-Kommando folgt die Länge der Hex-Paare der PDU-Zeichenkette ohne die beiden führenden Nullen. Das heißt, dass im obigen Beispiel 23 Hex-Paare an das Mobiltelefon gesendet werden, aber lediglich 22 angegeben werden müssen. Die eigentliche PDU-Zeichenkette beginnt mit „00“ und gibt die Länge der Short Massage Service Centre Nummer (SMSC) wieder. Die Längenangabe von null bedeutet, dass das Mobiltelefon selbst die eingespeicherte SMSC Nummer wählt. Mit Hilfe des zweiten Hex-Paares können verschiedene Optionen eingestellt werden. Die Angabe von „01“ bedeutet, dass die SMS an das SMSC geschickt wird. Das nächste Hex-Paar dient zum Zuordnen weiterer Nachrichten wie z. B. Berichten und ist die Referenznummer der Nachricht. Die Angabe von „00“ veranlasst dabei das Mobiltelefon dazu, die Referenznummer selbst zu wählen. Als nächstes folgen die Angabe der Länge der Empfängernummer, der Nummerntyp und entsprechend nachfolgend die Nummer. Die Länge der Empfängernummer wird durch das Hex-Paar „0B“ angegeben und ist somit 11 Zeichen lang. Das Hex-Paar „81“ gibt an, dass die nachfolgende Empfängernummer eine nationale Nummer ist. Bei der Angabe der Empfängernummer ist zu beachten, dass die einzelnen Ziffern der Paare der Nummer vertauscht werden. Aus der Nummer 01 62 12 34 56 7 wird dementsprechend 10 26 21 43 65 F7. Wie im Beispiel zu sehen, werden unvollständige Paare mit „F“ aufgefüllt. Die nächsten beiden Hex-Paare sind der so genannte Protocoll Identifier und das Data Coding Scheme. Hier genügt jeweils die Angabe von „00“. Es folgt die Angabe der Länge der zu sendende Nachricht, hier „0A“. Im Anschluss daran wird die eigentliche Nachricht als PDU codierter String angehängt. Die Vorgehensweise ist dabei in Tabelle 4-2 und Tabelle 4-3 dargestellt.

Tabelle 4-2 Vorgehen beim Umwandeln einer Zeichenkette in das PDU-Format (1)

ASCII Zeichen	H	a	l	l	o	
ASCII Hex	48	61	6C	6C	6F	20
8 Bit Binär	01001000	01100001	01101100	01101100	01101111	00100000
7 Bit Binär	1001000	1100001	1101100	1101100	1101111	0100000
8 Bit Binär PDU	11001000	00110000	10011011	11111101	00000110	01011101
8 Bit Hex PDU	C8	30	9B	FD	06	5D

Tabelle 4-3 Vorgehen beim Umwandeln einer Zeichenkette in das PDU-Format (2)

ASCII Zeichen	W	e	l	t
ASCII Hex	57	65	6C	74
8 Bit Binär	01010111	01100101	01101100	01110100
7 Bit Binär	1010111	1100101	1101100	1110100
8 Bit Binär PDU	11001011		01101100	00111010
8 Bit Hex PDU	CB		6C	3A

Bei der Umwandlung wird zunächst der ASCII-Text in eine 7Bit Darstellung überführt, indem das erste Bit gelöscht wird. Das bedeutet, dass nicht der gesamte ASCII-Zeichensatz zur Verfügung steht, sondern lediglich der in Anhang 6.4 dargestellte Zeichensatz. Als nächstes wird die 7Bit Darstellung wiederum in eine 8Bit Darstellung umgewandelt, indem beim ersten Zeichen das letzte Bit des nachfolgenden Zeichens am Anfang angefügt wird (siehe rote Markierungen in Tabelle 4-2 und Tabelle 4-3). Beim zweiten Zeichen werden entsprechend die letzten beiden Bits des nachfolgenden Zeichens angefügt. Dies wird solange fortgeführt, bis die kompletten 7Bit eines Zeichens an das vorhergehende angefügt worden und der Algorithmus von

Anfang an beginnt. Stehen keine Zeichen mehr zur Verfügung, so wird mit Nullen aufgefüllt, bis das Oktett vollständig ist (siehe blaue Markierungen Tabelle 4-3). Durch die Umwandlung von 8Bit ASCII Zeichen in eine PDU-Zeichenkette konnte im obigen Beispiel ein Zeichen bei der Übertragung eingespart werden. Die zu übertragende Nachricht „Hallo Welt“ wird damit durch „C8309BFD065DCB6C3A“ repräsentiert.

Der gesamte beschriebene Algorithmus ist als Programmablaufplan in Abbildung 4-12 und Abbildung 4-13 zu sehen.

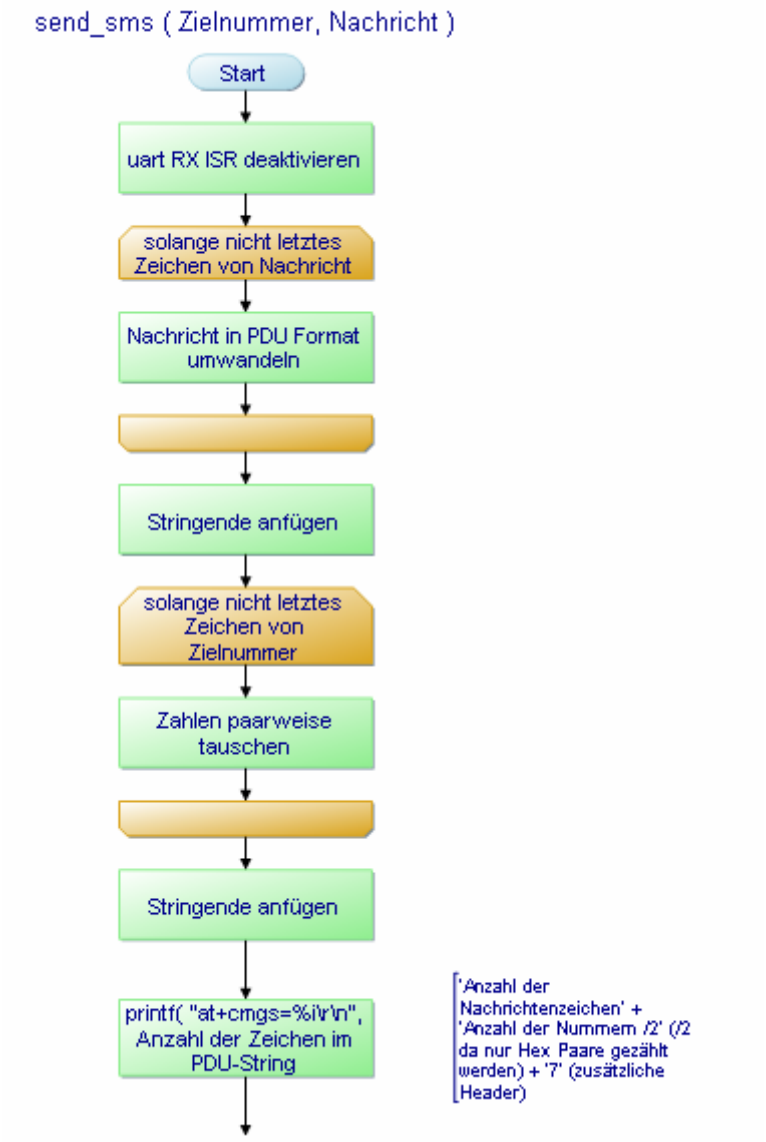

Abbildung 4-12 Übertragung einer Kurznachricht (1)

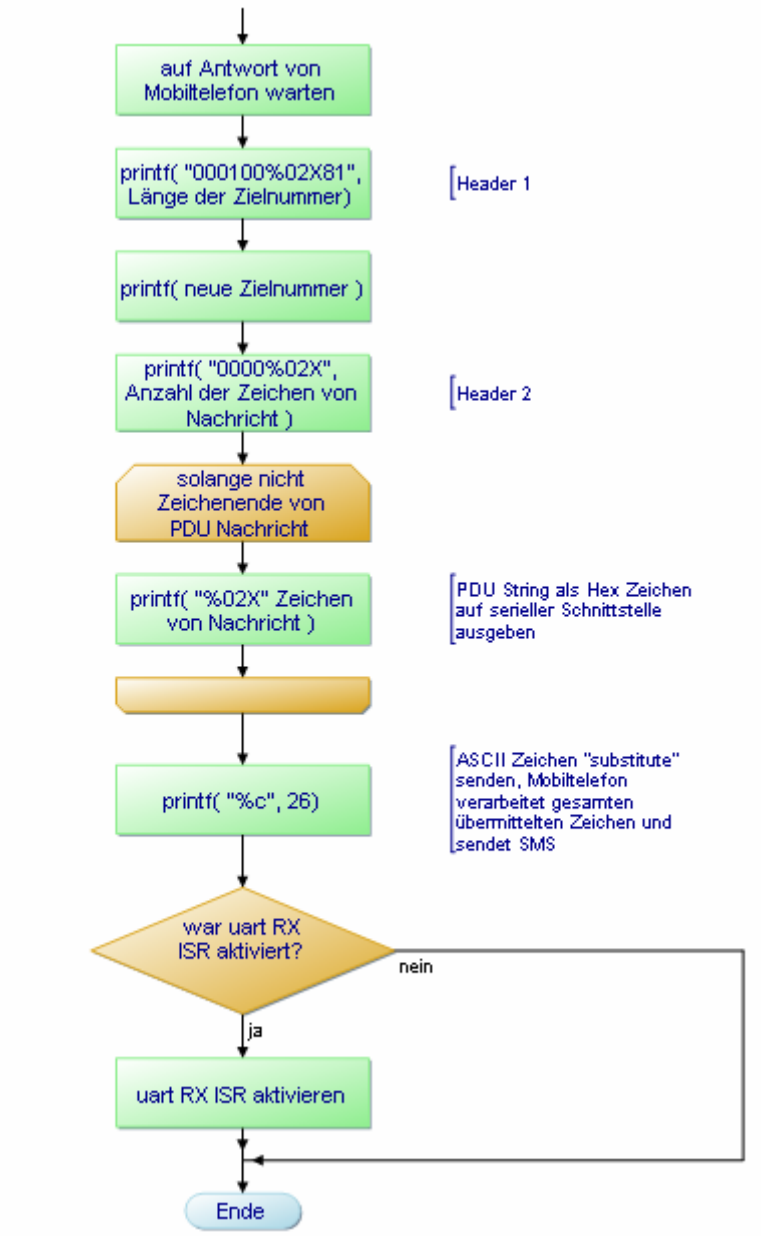

Abbildung 4-13 Übertragung einer Kurznachricht (2)

Zusätzlich zur Umsetzung des Algorithmus zum Versenden einer Kurznachricht im PDU-Format wurde ebenso eine Funktion zum Auslesen der Zeit und des Datums des Mobiltelefons entwickelt. Mit Hilfe des Befehls AT-Kommandos „AT+CCLK?“ sendet das angeschlossene Mobiltelefon das Datum und die eingestellte Zeit im Format "YYMMDD,hh:mm:ss“ zurück. Diese Ausgabe kann durch die vorhergehende Umleitung der Standard-Ein und Ausgabe mit der Funktion scanf() formatiert eingelesen und auf die entsprechenden Variablen geschrieben werden. Der Programmablaufplan ist in Abbildung 4-14 dargestellt.

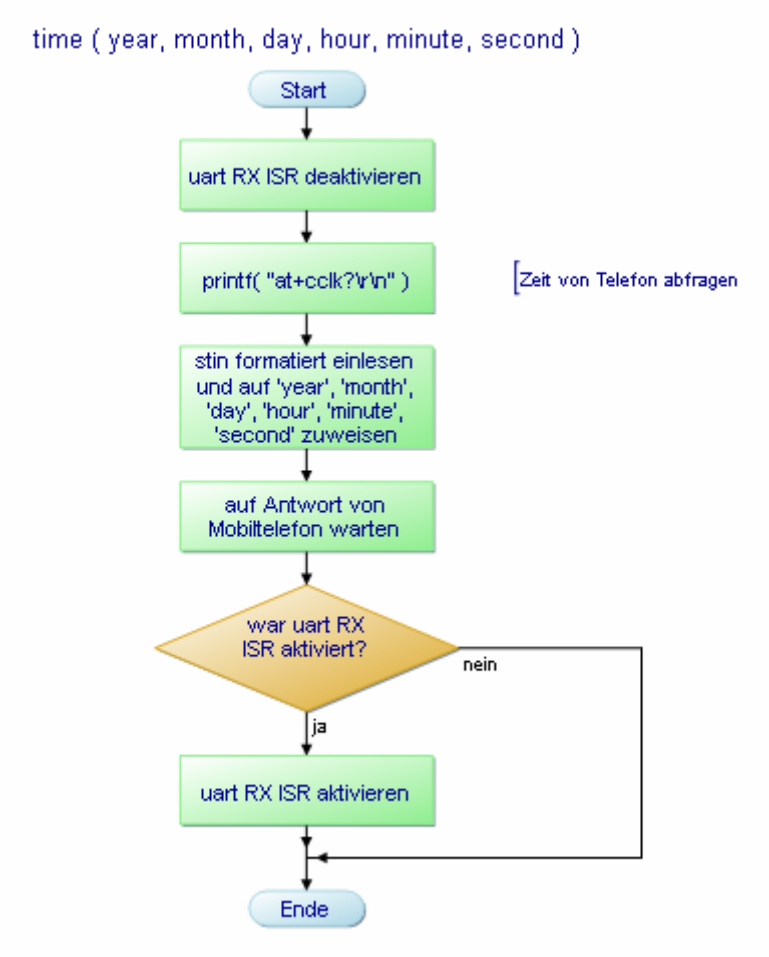

Abbildung 4-14 Programmablaufplan zum Auslesen von Datum und Zeit des Mobiltelefons