# KEN-TON DUAL RS-232 PAK

This ASCI board is memory mapped to addresses FF68 to FF6B (Hex).

The 6551 uses 4 memory addresses for the following functions:

| --PAK #1-- /T2 | --PAK #1-- /T3 | --PAK #2-- /T4 | --PAK #2-- /T5 | Write | Read |
|---|---|---|---|---|---|
| [FF68] | [FF6C] | [FF60] | [FF64] | Transmit data | Receiver Data |
| [FF69] | [FF6D] | [FF61] | [FF65] | Register | Register |
| [FF6A] | [FF6E] | [FF62] | [FF66] | Soft Reset | Status Register |
| [FF6B] | [FF6F] | [FF63] | [FF67] | Command Register | |
| J1=LO | J1=HI | J1=LO | J1=H1 | Control Register | |

This table shows that only the Command and Control registers are bidirectional. The Soft Reset operation clears the 6551 internal registers but does not cause any data transfer. Therefore, the data written is a "don't care." A RES* from the bus will clear all registers in the 6551 while a Soft Reset will disable interrupts and transmitter, turn off echo mode, and clear the Overrun Error flag only.

## Dimensions

| | |
|---|---|
| **Length** | 6.6" (16.5 cm) |
| **Width** | 4.4" (11 cm) |
| **Height** | 1" (2.5 cm) |
| **Weight** | 6.5 oz. (182 g) |

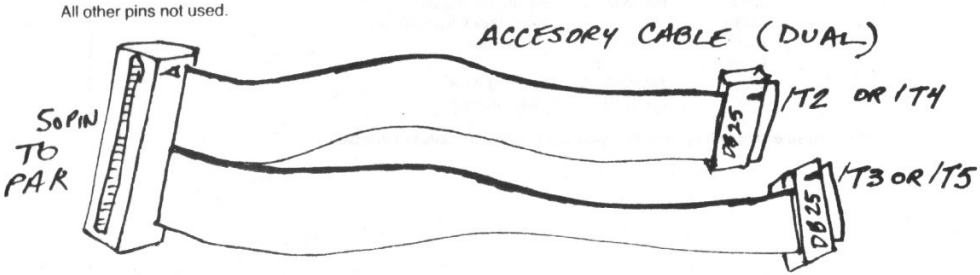
## Environmentals

| | |
|---|---|
| **Temperature** | |
| Operating | 41° to 104° F. (5° to 40° C.) |
| Storage | – 40° to 160° F. ( – 40° to 71° C.) |
| **Humidity** | |
| Operating | 40% to 80% non-condensing |
| Storage | 20% to 90% non-condensing |

## RS-232C Pin Description

| Pin Number | Signal |
|---|---|
| 1 | Frame Ground |
| 2 | Transmit Data |
| 3 | Receive Data |
| 4 | Request to Send |
| 5 | Clear to Send |
| 6 | Data Set Ready |
| 7 | Signal Ground |
| 8 | Carrier Detect |
| 20 | Data Terminal Ready |

All other pins not used.



# Programming Examples (Machine-Language)

## Desired Configuration

**Baud Rate** = 300, Word Length = 7 Bits, Stop = 2 Bits, Parity = Even

| Instruction | | | Comment |
|---|---|---|---|
| **Control Register** | | | |
| | CLRA | | |
| | ORA | #$80 | 2 Stop Bits |
| | ORA | #$20 | 7-Bit Word |
| | ORA | #$10 | Receiver Clock = Baud Rate Gen. |
| | ORA | #$06 | 300 baud rate |
| | STA | $FF6B | Write it to Control Register |
| **Command Register** | | | |
| | CLRA | | |
| | ORA | #$60 | Even Parity |
| | ORA | #$00 | MODE = Normal |
| | ORA | #$08 | Tx. Int. Disabled, RTS Active Tx. On |
| | ORA | #$01 | IRQ Int. Disabled |
| | ORA | #$01 | Rx Enabled and All Int. |
| | STA | $FF6A | Write it to Command Register |

**Status Register**

| | | | |
|---|---|---|---|
| **To Check Modem Status (DSR):** | | | |
| CKMDM | LDA | $FF6A | Load Command Byte |
| | ORA | #$01 | Insure that DTR Bit is active |
| | STA | $FF6A | Output to Command Register |
| | LDA | $FF69 | Read Status Register |
| | ANDA | #$40 | Check DSR Status |
| | BNE | CKMDM | Not Ready |
| | BRA | MREADY | Modem is ready (powered-up) |
| **To Check the Carrier Status (DCD):** | | | |
| | LDA | $FF69 | |
| | ANDA | #$20 | |
| | BNE | CAROFF | Carrier is not present |
| | BRA | CARON | Carrier is present |
| **To Transmit a Data Byte:** | | | |
| TXDT | LDA | $FF69 | Check Tx-Register Status |
| | ANDA | #$10 | If empty or not |
| | BEQ | TXDT | Not empty, wait Tx Data |
| | LDA | DATA | Get data byte from RAM |
| | STA | $FF68 | Write it to data register |

| | | | |
|---|---|---|---|
| **To receive a data byte:** | | | |
| RXDT | LDA | $FF69 | Read Status Register |
| | TFR | A,B | Save Status |
| | ANDA | #$08 | Check if received any data |
| | BEQ | RXDT | Not received yet |
| | ANDB | $#07 | Received, check any error |
| | BEQ | RXDATA | No error, good data |
| | LSRB | | Error, check type of error |
| | BCS | PARERR | Parity error |
| | LSRB | | |
| | BCS | FRAERR | Framing error |
| | BRA | OVRERR | Overrun error |

***Note:** These example programs are applicable when interrupts are disabled.

# Programming Examples (BASIC)

## Desired Configuration:

Baud Rate = 300, Word Length = 7 Bits, Stop = 2 Bits, Even Parity

| Instruction | Comment |
|---|---|
| **To program the Control Register:** | |
| `10 X = 182` | This is a B6(HEX)<br>300 baud, 7 bit word, 2 stop bits |
| `20 POKE 65387,X` | Write to Control Register FF6B (HEX) |
| **To program the Command Register:** | |
| `30 Y = 107` | This is 6B (HEX); Even Parity, Normal Receive, Trans. Int. Disabled, RTS Active, IRQ Disabled, Receiver Enabled. |
| `40 POKE 65386,Y` | Write to Command Register FF6A (HEX) |
| **To check Modem status:** | |
| `50 A = PEEK(65386)` | Get current command Byte |
| `60 A = A OR 1` | OR in DTR enable BIT |
| `65 B = POKE 65386,A` | Output new Command Byte |
| `70 B = PEEK(65385)` | Read Status |
| `75 B = B AND 64` | Check only DSR Status |
| `80 IF B = 0 THEN 100` | If B = 0, Modem is ready |
| `90 PRINT"MODEM IS NOT READY"` | B>0, Modem is not ready |
| `95 END` | |
| `100 PRINT"MODEM IS READY"` | |

| | |
|---|---|
| **To Check Carrier Status:** | |
| `110 Z = PEEK(65385)` | Mask off all but DCD Status |
| `120 Z = Z AND 32` | If Z = 0, then we have carrier; otherwise, we do not |
| `130 IF Z = 0 THEN 160` | |
| `140 PRINT "NO CARRIER"` | |
| `150 END` | |
| `160 PRINT "GOT CARRIER!"` | |
| **To Transmit a Data Byte:** | |
| `200 A = PEEK(65385)` | FF69(HEX) |
| `210 A = A AND 16` | Transmit Holding Register empty? |
| `220 IF A= 0 THEN 200` | Not empty |
| `230 POKE 65384,D` | D has data byte |
| **To Receive a Data Byte:** | |
| `300 A = PEEK(65385)` | Read status register |
| `310 B = A AND 8` | Receive register full? |
| `320 IF B = 0 THEN 300` | No data yet |
| `330 E = A AND 7` | Check for errors |
| `340 IF E = 0 THEN 400` | No errors |
| `350 IF E = 1 THEN 370` | Parity error |
| `355 IF E = 2 THEN 380` | Framing error |
| `360 PRINT "OVERRUN ERROR!"` | Must be overrun error |
| `365 END` | |
| `370 PRINT "PARITY ERROR!"` | |
| `375 END` | |
| `380 PRINT "FRAMING ERROR!"` | |
| `390 END` | |
| `400 D = PEEK (65384)` | |
| `410 PRINT "BYTE RECEIVED WAS";D` | Got data byte in D |

**CONTROL REGISTER**
**The Control Register is used to select the desired mode for the SY6551.**

**The word length, number of stop bits, and clock controls are all determined by the Control Register.**

CONTROL REGISTER

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|

STOP BITS

0 = 1 Stop Bit
1 = 2 Stop Bits
1 Stop Bit if Word Length = 8 Bits and Parity*
1½ Stop Bits if Word Length = 5 Bits and No Parity.

WORD LENGTH

| BIT 6 | BIT 5 | DATA WORD LENGTH |
|---|---|---|
| 0 | 0 | 8 |
| 0 | 1 | 7 |
| 1 | 0 | 6 |
| 1 | 1 | 5 |

RECEIVER CLOCK SOURCE

0 = External Receiver Clock
1 = Baud Rate Generator

*This allows for 9-bit transmission (8 data bits plus parity).

BAUD RATE GENERATOR

| | | | | |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 16x EXTERNAL CLOCK |
| 0 | 0 | 0 | 1 | 50 BAUD |
| 0 | 0 | 1 | 0 | 75 |
| 0 | 0 | 1 | 1 | 109.92 |
| 0 | 1 | 0 | 0 | 134.58 |
| 0 | 1 | 0 | 1 | 150 |
| 0 | 1 | 1 | 0 | 300 |
| 0 | 1 | 1 | 1 | 600 |
| 1 | 0 | 0 | 0 | 1200 |
| 1 | 0 | 0 | 1 | 1800 |
| 1 | 0 | 1 | 0 | 2400 |
| 1 | 0 | 1 | 1 | 3600 |
| 1 | 1 | 0 | 0 | 4800 |
| 1 | 1 | 0 | 1 | 7200 |
| 1 | 1 | 1 | 0 | 9600 |
| 1 | 1 | 1 | 1 | 19.200 |

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| HARDWARE RESET | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| PROGRAM RESET | – | – | – | – | – | – | – | – |

Control Register Format

**COMMAND REGISTER**
**The Command Register is used to control Specific Transmit Receive functions.**

COMMAND REGISTER

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|

PARITY CHECK CONTROLS

| BIT 7 | BIT 6 | BIT 5 | OPERATION |
|---|---|---|---|
| – | – | 0 | Parity Disabled - No Parity Bit Generated - No Parity Bit Received |
| 0 | 0 | 1 | Odd Parity Receiver and Transmitter |
| 0 | 1 | 1 | Even Parity Receiver and Transmitter |
| 1 | 0 | 1 | Mark Parity Bit Transmitted, Parity Check Disabled |
| 1 | 1 | 1 | Space Parity Bit Transmitted, Parity Check Disabled |

NORMAL ECHO MODE FOR RECEIVER

0 = Normal
1 = Echo (Bits 2 and 3 must be "0")

DATA TERMINAL READY

0 = Disable Receiver and All Interrupts ($\overline{DTR}$ high)
1 = Enable Receiver and All Interrupts ($\overline{DTR}$ low)

RECEIVER INTERRUPT ENABLE

0 = $\overline{IRQ}$ Interrupt Enabled from Bit 3 of Status Register
1 = $\overline{IRQ}$ Interrupt Disabled

TRANSMITTER CONTROLS

| BIT 3 | BIT 2 | TRANSMIT INTERRUPT | $\overline{RTS}$ LEVEL | TRANSMITTER |
|---|---|---|---|---|
| 0 | 0 | Disabled | High | Off |
| 0 | 1 | Enabled | Low | On |
| 1 | 0 | Disabled | Low | On |
| 1 | 1 | Disabled | Low | Transmit BRK |

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| HARDWARE RESET | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| PROGRAM RESET | – | – | – | 0 | 0 | 0 | 0 | |

Command Register Format

**STATUS REGISTER**
**The Status Register is used to indicate to the processor the status of various SY6551 functions.**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|

| STATUS | SET BY | CLEARED BY |
|---|---|---|
| Parity Error* | 0 = No Error<br>1 = Error | Self Clearing** |
| Framing Error* | 0 = No Error<br>1 = Error | Self Clearing** |
| Overrun* | 0 = No Error<br>1 = Error | Self Clearing** |
| Receive Data Register Full | 0 = Not Full<br>1 = Full | Read Receive Data Register |
| Transmit Data Register Empty | 0 = Not Empty<br>1 = Empty | Write Transmit Data Register |
| $\overline{DCD}$ | 0 = $\overline{DCD}$ Low<br>1 = $\overline{DCD}$ High | Net Reusable Reflects $\overline{DCD}$ State |
| $\overline{DSR}$ | 0 = $\overline{DSR}$ Low<br>1 = $\overline{DSR}$ Low | Net Reusable Reflects $\overline{DSR}$ State |
| IRQ | 0 = No Interrupt<br>1 = Interrupt | Read Status Register |

*NO INTERRUPT GENERATED FOR THESE CONDITIONS.
**CLEARED AUTOMATICALLY AFTER A READ OF RDR AND THE NEXT ERROR-FREE RECEIPT OF DATA.

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| HARDWARE RESET | 0 | – | – | 1 | 0 | 0 | 0 | 0 |
| PROGRAM RESET | – | – | – | – | – | 0 | – | – |