Serial control
Documentation
Hardware V2.13* – Software Vx.33F*

Date: April, 2012

Version: 20120418
Status: Final

Owner: AMT Netherlands

Document history

Revision	Date	Major modifications	Chapter	Ref. number decision form
20110615	Jun, 2011	Init document. Split from master document	All	
20111207	Dec, 2011	Added serial communication input values for OPS & THR	2.2.2, 2.2.3	
20120118	Jan, 2012	Changed offset throttle from 16 to 32		
20120306	Mar, 2012	Added sequence of sending serial data stream		
20120410	Apr, 2012	Update paragraph 2.2.1 and added paragraph 2.3.2	2.2.1, 2.3.2	
20120418	Apr, 2012	Update paragraph 2.2.5	2.2.5	

***And higher versions**

Table of contents

1 Introduction

1.1 Objective of this document

This document contains the information of the serial communication implemented in the ECU hardware .
The target group of this document is AMT-Netherlands and third parties.

1.2 Cohesion of documents

Together with the Engine Control Unit (ECU) specifications and the communication specification a complete set of specification documents is formed. Development is and will be based upon this set of documents.

1.3 Scope of this document

This document is suited for ECU of AMT Netherlands. Wherever possible any deviations for other than AMT-Netherlands developments will be indicated.

1.4 About this document

This document starts with general requirements for the system and is followed by sections that deal with the signals and features of the ECU.

1.5 Links to other documents

The following sources serve as input for this document:

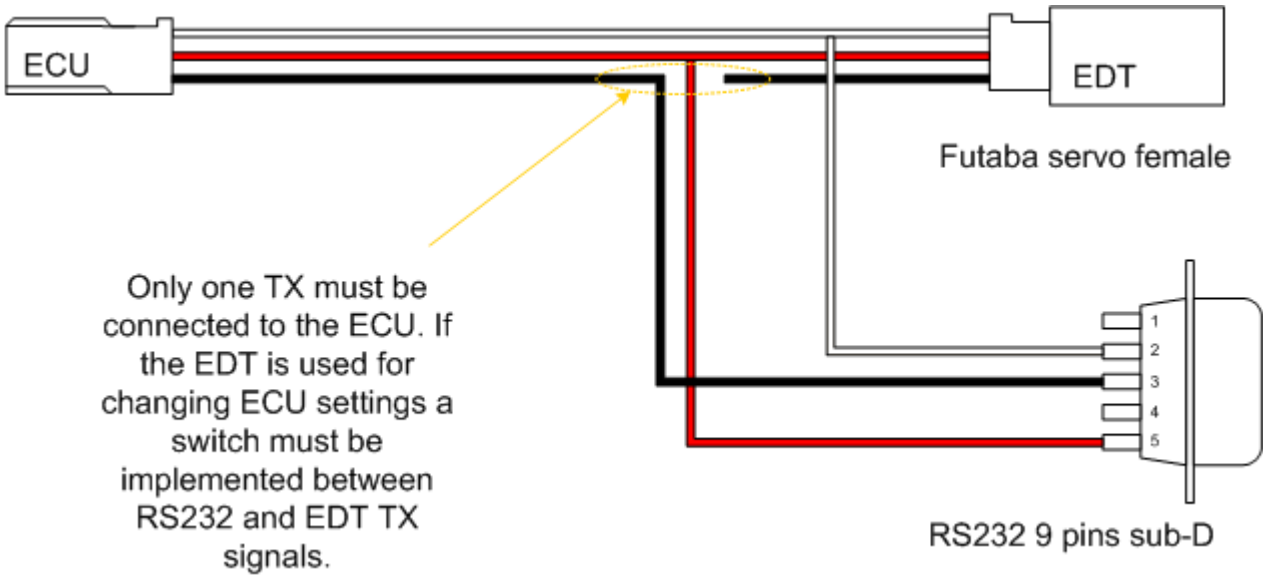
- AMT requirements ..... AR
- Specification: Hardware ECU ..... SH
- Specification: Software ECU ..... SS

2 Serial control

2.1 General

Besides receiver pulses and analog control a third option is possible, via a serial protocol. The EDT connection is used to send data to the ECU with OPS and THR information.
For safety reasons a constant stream of data is required. If this stream is missing for a period, adjustable by a parameter, the turbine will be stopped.

To interface the EDT and serial communication at the same time next wiring harness is needed.



If an EDT is used which can alter the settings of the ECU the TX of the EDT can't be connected directly to the ECU. A collision will take place between the TX signals of the RS232 connection and from the EDT. If both connections are preferred a switch must be implemented for switching between TX signal of the RS232 connection and the TX signal of the EDT.

2.2 EDT data

The ECU reports its status and condition via a serial protocol. This serial protocol is based on the industrial standard RS232.
After power up the ECU transmitted the software version, the software date and a number of settings. After this, the normal serial data is transmitted.

2.2.1 General specifications

Item	Description
Level	Standard RS232 level -12V to 12V
Baud rate	Standard the ECU setting is 2400. This gives an average of 48 bytes per second. Other settings are 2400-4800-9600-19200-38400-57600-115200.
Protocol setup	8 data bits, no parity, 1 stop-bit
Data stream*	0xFF,{data1},{data2},{data3},{data4},{data5}

*Value of the data bytes 1 to 5 is always between 0 and 0xFE (254). After 10 data sets of normal information an alternating data set is send.

2.2.2 Normal information data set

Byte	Unit	Description
Leader		Value always 0xFF (255)
Data 1 (Status)		This data byte describes the state of the ECU and which type of engine is installed. If B2, B1 and B0 are zero (low) then the bits B7 to B3 will indicated which engine is installed.

Bits								Description
7	6	5	4	3	2	1	0	
0	0	0	0	0	0	0	0	Error (See Error information set)
X	X	X	X	X	0	0	1	Operator Switch in Emergency stop mode
X	X	X	X	X	0	1	0	Operator Switch in Auto stop mode
X	X	X	X	X	1	0	0	Operator Switch in Running mode
0	0	0	0	0	0	1	1	PEGASUS engine ID (older versions)
0	0	0	0	0	1	1	0	OLYMPUS engine ID (older versions)
0	0	0	0	0	1	1	1	MERCURY engine ID (older versions)
0	0	0	0	1	0	0	0	MERCURY engine ID
0	0	0	1	0	0	0	0	PEGASUS engine ID
0	0	0	1	1	0	0	0	OLYMPUS engine ID
0	0	1	0	0	0	0	0	TITAN engine ID
0	0	1	0	1	0	0	0	NIKE engine ID
X	X	X	X	1	S	S	S	Start clearance
X	X	X	1	X	S	S	S	Starting
X	X	1	X	X	S	S	S	Started up
X	1	X	X	X	S	S	S	Idle calibration
X	1	1	X	X	S	S	S	Full operation running turbine
1	X	X	X	X	S	S	S	Maximum RPM reached

X = don't care 1 or 0
S = state as in Operator Switch (001 = emer. Stop, 010 = auto stop, 100 = run)

Data 2 (RPM value)	RPM	Motor type : PEGASUS, OLYMPUS, TITAN, NIKE RPM = value * 500 Motor type : MERCURY RPM = value * 700
Data 3 (EGT value)	˚C	EGT = value * 4.6 – 50
Data 4 (Throttle setting)	%	THR = value / 2
Data 5 (Vout value)	Volt	VOUT = value * 6.25 / 255 (Engine ID: MERCURY, PEGASUS)
		VOUT = value * 8.30 / 255 (Engine ID: OLYMPUS, TITAN, NIKE)

2.2.3 Error information data set

Byte	Unit	Description
Leader		Value always 0xFF (255)
Data 1 (Status)		This data byte describes the state of the ECU and when it's in the error mode this value is ZERO.
Data 2 (Error code)		RPM value is overruled with the error code of the ECU. Next table gives the relation between error code and the bits which are set.

Bits								Description
7	6	5	4	3	2	1	0	
X	X	X	X	X	X	X	1	rpm low
X	X	X	X	X	X	1	X	switch channel not present
X	X	X	X	X	1	X	X	throttle channel not present
X	X	X	X	1	X	X	X	EGT error
X	X	X	1	X	X	X	X	rpm high
X	X	1	X	X	X	X	X	supply low
X	1	X	X	X	X	X	X	supply low for Auto Start System
1	X	X	X	X	X	X	X	

X = don't care 1 or 0

Data 3 (EGT value)	˚C	EGT = value * 4.6 – 50
Data 4 (Throttle setting)	%	THR = value / 2
Data 5 (Vout value)	Volt	VOUT = value * 6.25 / 255 (Engine ID: MERCURY , PEGASUS)
		VOUT = value * 8.30 / 255 (Engine ID: OLYMPUS, TITAN, NIKE)

2.2.4 Alternate information data set

Byte	Unit	Description
Leader		Value always 0xFF (255)
Data 1 (Engine ID)		If B2, B1 and B0 are zero (low) then the bits B7 to B3 will indicated which engine is installed.

Bits								Description
7	6	5	4	3	2	1	0	
0	0	0	0	0	0	1	1	PEGASUS engine ID (older versions)
0	0	0	0	0	1	1	0	OLYMPUS engine ID (older versions)
0	0	0	0	0	1	1	1	MERCURY engine ID (older versions)
0	0	0	0	1	0	0	0	MERCURY engine ID
0	0	0	1	0	0	0	0	PEGASUS engine ID
0	0	0	1	1	0	0	0	OLYMPUS engine ID
0	0	1	0	0	0	0	0	TITAN engine ID
0	0	1	0	1	0	0	0	NIKE engine ID

Data 2 (idle voltage)	Volt	PWOMIN = value * 6.25 /255 (Engine ID: MERCURY , PEGASUS)
		PWOMIN = value * 8.30 / 255 (Engine ID: OLYMPUS, TITAN, NIKE)
Data 3 (max rpm voltage)	Volt	PWOMAX = value * 6.25 / 255 (Engine ID: MERCURY , PEGASUS)
		PWOMAX = value * 8.30 / 255 (Engine ID: OLYMPUS, TITAN, NIKE)
Data 4 (battery voltage)	Volt	SUPLVAL = value * 10.0 / 255 (Engine ID: MERCURY , PEGASUS)
		SUPLVA L= value * 10.0 / 255 (Engine ID: OLYMPUS, TITAN)
Data 5 (ext. batt. voltage)	Volt	Vout= 7 + (value * 6.25 / 255) (Engine ID: MERCURY , PEGASUS)
		Vout= 7+ (value * 9.30 / 255) (Engine ID: OLYMPUS, TITAN, NIKE)

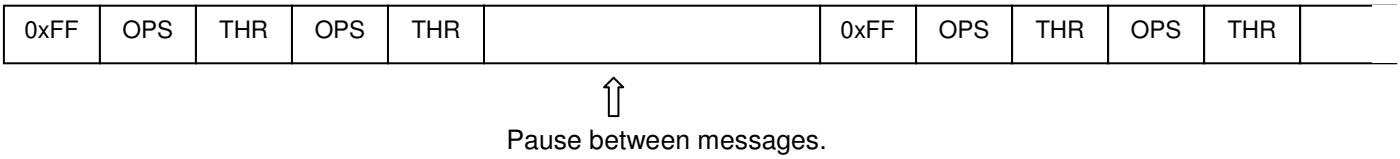
2.2.5 ECU set-up data set

Byte	Unit	Description
Leader		Value always 0xFF (255)
Data 1 (ECU set-up)		Value always 0x05 (5)
Data 2 (high byte)*	ms	Input pulse width information from the switch channel
Data 3 (low byte)*		Pulse width minimum: PWmin = (((256 * PWTH)+PWTL)-62464)/1000
Data 4 (pulse difference)*	ms	PWmax = PWmin + PWIDIFF * 16
Data 5…17		A string of 12 bytes is transmitted (ASCII) which containing the software version and the production date. Format: "X.NN YYMMDD " (old) “XX.NN YYMMDD “ (new) Software version and engine type. 1.NN: Pegasus engine (obsolete) 2.NN: Pegasus engine (obsolete) 3.NN: Olympus engine 4.NN: Pegasus engine 5.NN: Mercury engine 6.NN: Olympus engine 7.NN: Pegasus engine 8.NN: Mercury engine 9.NN: Olympus engine 10.NN: Titan engine 12.NN: Nike engine

*Only to be used when Radio Controlled pulse method is switch on as input value

2.3 Serial communication implementation

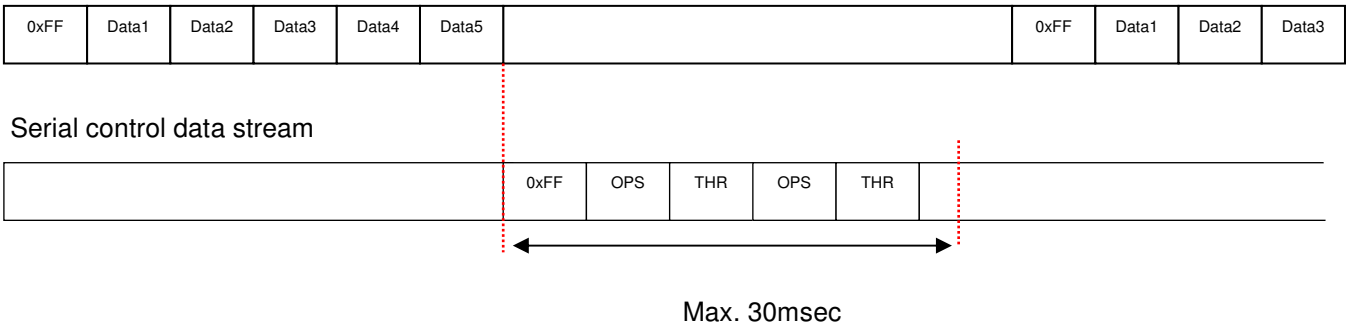
2.3.1 Protocol



The message will start with a 255 byte value to indicate that new OPS (switch) and THR (throttle) info is coming. OPS and THR will be sending twice and the ECU software will determine if both bytes of the OPS and THR are the same. When these bytes are not the same the message is ignored and the error counter increased.

To make sure data will be excepted without errors next implementation must be guaranteed. This means that a serial control data stream must be send directly after last byte of EDT data stream is received.

EDT data stream



Calculation:

- 2400Baud = 2400bps → 2400*0.030 = 72 bits per 30msec.
- Databyte has 1*startbit, 8*databits, no parity and 1*stopbit so in total 10 bits.
- Time per byte: (1 / 2400) * 10 = 4,16msec.
- Time serial data stream: 5(data bytes) x 10bits = 50 bits or 5(data bytes) * 4,16 = 20,8msec.

So a data stream at 2400Baud fits within the max. time frame of 30msec.

2.3.2 Missing serial data

In case of a missing serial data stream to the ECU, the ECU will stop the engine and generate a “switch fail” and “”throttle fail” when the serial delay time (address 192) is elapsed.

The “standard” setting of address 192 is 50, this value of 50 gives a delay time equal to 2.8 seconds at a “running engine”. When the engine is not running this value of 50 gives a delay time equal to 5.6 seconds.

2.3.3 General specifications

Item	Description
Level	Standard RS232 level -12V to 12V
Baud rate	Standard the ECU setting is 2400. This gives an average of 48 bytes per second. Other settings are 2400-4800-9600-19200-38400-57600-115200.
Protocol setup	8 data bits, no parity, 1 stop-bit
Data stream*	0xFF,{data1},{data2},{data3},{data4}

*Value of the data bytes 1 to 4 is always between 0 and 0xFE (254).

2.3.4 Engine control data set

<table>
<tr><th>Byte</th><th>Unit</th><th>Description</th></tr>
<tr><td>Leader</td><td></td><td>Value always 0xFF (255)</td></tr>
<tr><td>Data 1 (OPS Status)</td><td></td><td>This data byte describes the state of the Operation Switch (OPS).
<table>
<tr><th colspan="8">Bits</th><th>Description</th></tr>
<tr><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th><th></th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>Operator Switch in Emergency stop mode</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>Operator Switch in Auto stop mode</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>Operator Switch in Running mode</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>CTF (special functions)*</td></tr>
</table></td></tr>
<tr><td>Data 2 (THR value)</td><td>%</td><td>This data byte describes the throttle setting.

Data2 = (THR * 2) + 32

Example: 50% → (50 * 2) + 32 = 132</td></tr>
<tr><td>Data 3 (OPS status)</td><td></td><td>See Data 1</td></tr>
<tr><td>Data 4 (THR value)</td><td>%</td><td>See Data 2</td></tr>
</table>

*If OPS value is 145 (CTF) during running mode, engine will stop after serial delay time (address 192) is elapsed.

Appendix A Glossary

AMT	Advanced Micro Turbine the Netherlands
AR	AMT requirements
ECU	Engine Control Unit
EGT	Exhaust Gas Temperature
EMC	Electro Magnetic Compatibility
ESD	Electro Static Discharge.
FET	Field Effect Transistor
HAL	Hardware abstraction layer
LCD	Liquid Crystal Display
LED	Light Emitting Diode
MC	Machine Controller
MTBF	Mean Time Between Failures
MTTF	Mean Time To Failure
OH	Old ECU hardware specifications
OPS	Operational switch
OS	Old ECU software specifications
PCB	Printed Circuit Board
PLL	Phase Locked Loop.
RPM	Revolutions per minute
Rx	Receive
Tbd	Too be defined
THR	Throttle
TSOP	Thin Small Outline Package
Tx	Transmit
UART	Universal Asynchronous Receiver Transmitter
UI	User Interface
Vout	Voltage on fuel pump output

