Defence Research and Development Canada | Recherche et développement pour la défense Canada

DEFENCE R&D DÉFENSE

Documentation of the Carleton University Conducted Energy Weapons (CEW) Test Analysis Software

Andy Adler
Owen Marsh
David P. Dawson

Prepared By:
Carleton University
1125 Colonel By Drive
Ottawa, Ontario
K1A 5B6
Systems and Computer Engineering
Contractor's Document Number: Contractor's Document Number:
Contract Project Manager: Andy Adler
PWGSC Contract Number: PWGSC Contract Number: W7714-4500833623
CSA: Donna Wood, Project Manager CEWSI, 613-943-2472

Defence R&D Canada – Centre for Security Science

Contract Report

DRDC CSS CR 2011-18

October 2011

Canada

Documentation of the Carleton University Conducted Energy Weapons (CEW) Test Analysis Software

Andy Adler
Owen Marsh
David P. Dawson

Prepared By:
Carleton University
1125 Colonel By Drive
Ottawa, Ontario
K1A 5B6
Systems and Computer Engineering
Contractor's Document Number:
Contract Project Manager: Andy Adler
PWGSC Contract Number: W7714-4500833623
CSA: Donna Wood, Project Manager CEWSI, 613-943-2472

Defence R&D Canada – Centre for Security science

Contract Report
DRDC CSS CR 2011-18
October 2011

Principal Author

Original signed by Andy Adler

Andy Adler

Canadian Research Chair

Approved by

Original signed by Donna Wood

Donna Wood

Project Manager, CEWSI

Approved for release by

Original signed by Mark Williamson

Mark Williamson

Chair DRDC Document Review Panel

This work was done for the Conducted Energy Weapons Strategic Initiative (CEWSI), project number 32bj, a project funded by the Canadian Police Research Centre (CPRC) and managed by Defence Research and Development Canada (DRDC) under the Centre for Security Science (CSS).

Abstract

This report describes the test analysis software developed at Carleton University (CEW) that was written to support the work on the Conducted Energy Weapons (CEW) Test Procedure, version 1.1 (published 31 July 2010). This report documents the software, its structure, its business rules, and it identifies any data elements required to fully support the published test procedure. This report is designed to support the Conducted Energy Weapons Strategic Initiative (CEWSI), which plans to test a number of CEWs and do analysis on legacy test data with the objective of better understanding and interpreting the performance of the devices, with a goal of developing an updated CEW Test Procedure and providing advice to Canadian Law Enforcement on these devices.

Résumé

Le présent rapport décrit le logiciel d’analyse d’essai mis au point par l’université Carleton (AI) qui a été élaboré pour appuyer les travaux menés sur les procédures de déroulement des essais sur les armes à impulsions (AI), version 1.1 (publiées le 31 juillet 2010). Le présent rapport décrit le logiciel, sa structure et ses règles administratives et il répertorie également tous les éléments de données nécessaires pour appuyer entièrement les procédures d’essai publiées. Le présent rapport est conçu pour appuyer l’Initiative stratégique sur les armes à impulsions (ISAI), qui prévoit la mise à l’essai d’un certain nombre d’AI et de faire l’analyse des données d’essai des anciens systèmes dans le but de mieux comprendre et de mieux interpréter le comportement de ces dispositifs, avec comme objectif de mettre au point de nouvelles procédures d’essai mises à jour des AI et de pouvoir conseiller les organismes canadiens chargés de l’application de la loi sur ces dispositifs.

This page intentionally left blank.

Executive summary

Documentation of the Carleton University Conducted Energy Weapons (CEW) Test Analysis Software

Andy Adler; Owen Marsh; David P. Dawson; DRDC CSS CR 2011-18; Defence R&D Canada – CSS; October 2011.

Introduction or background: There is a need for reliable uniform testing of Conducted Energy Weapons (CEWs) independent of the manufacturer, as identified by several studies including the Braidwood Commission report, the Report of the Standing Committee on Public Safety and National Security of the Conducted Energy Weapon. Recognizing this need, a team at Carleton University (lead by Professor A. Adler and Mr. D.P. Dawson) began a research program to develop techniques and data analysis methods for data from CEWs. This research has led to: 1) development of CEW data acquisition and test systems, 2) Development of software to analyse CEW, 3) organizing two Canadian CEW workshops, and, based on consensus at the workshop, 4) leading of a team to write a CEW Test Procedure (version 1.1, July 31, 2010).

Results: This report describes the test analysis software developed at Carleton University (CEW) that was written to support the work on the Conducted Energy Weapons (CEW) Test Procedure. This report documents the software, its structure, its business rules, and it identifies any data elements required to fully support the published test procedure

Significance: This report is designed to support the Conducted Energy Weapons Strategic Initiative (CEWSI), which plans to test a number of CEWs and do analysis on legacy test data with the objective of better understanding and interpreting the performance of the devices.

Future plans: CEWSI plans to use this report to further the goal of developing an updated CEW Test Procedure and providing advice to Canadian Law Enforcement on these devices.

Sommaire

Documentation of the Carleton University Conducted Energy Weapons (CEW) Test Analysis Software: Documentation relative au logiciel d’analyse d’essai élaboré par l’université Carleton pour les armes à impulsions (AI)

Andy Adler; Owen Marsh; David P. Dawson; DRDC CSS CR 2011-18; R & D pour la défense Canada – CSS; octobre 2011.

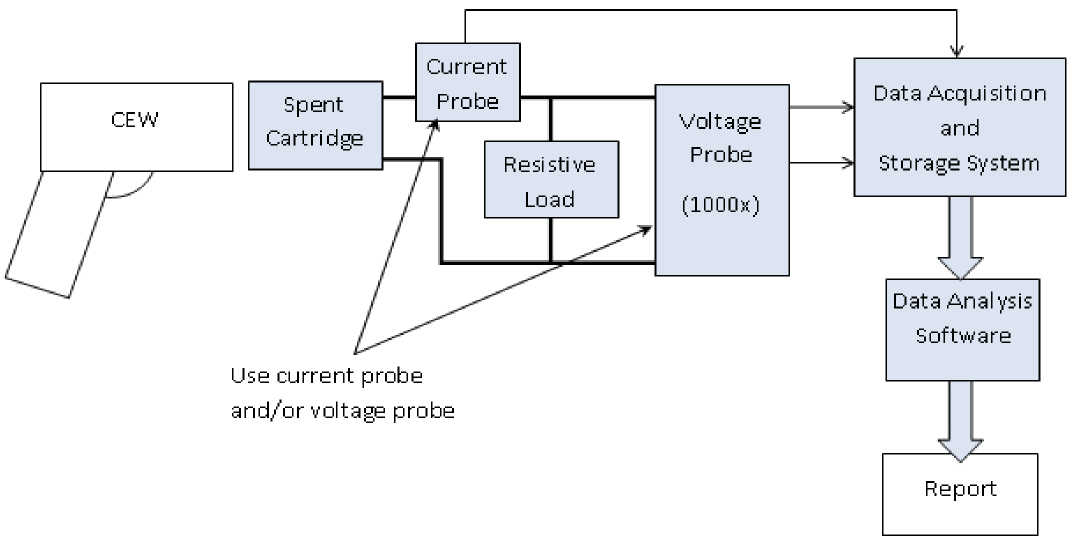
Introduction ou contexte: Tel que cela a été révélé par plusieurs études sur les armes à impulsions (AI), notamment le rapport de la Commission Braidwood et le rapport du Comité permanent de la sécurité publique et nationale, il est nécessaire de disposer de moyens d’évaluation des armes à impulsions (AI) qui soient uniformes et fiables; ces moyens d’évaluation doivent aussi être dissociés du fabricant de ces armes. Conscient de ce besoin, une équipe de l’université Carleton (dirigée par le Professeur A. Adler et par M. D.P. Dawson) a entrepris un programme de recherche visant à mettre au point des techniques et des méthodes d’analyse des données pour les AI. Cette recherche a mené à : 1) la mise au point de systèmes d’acquisition de données et d’essai pour les AI, 2) l’élaboration d’un logiciel permettant d’analyser les AI, 3) l’organisation de deux ateliers canadiens sur les AI et, en fonction du consensus établi lors de l’un de ces ateliers, 4) la création d’une équipe ayant pour but de rédiger des procédures de mise à l’essai des AI (version 1.1, 31 juillet 2010).

Résultats: Le présent rapport décrit le logiciel de mise à l’essai conçu à l’université Carleton (AI) afin d’appuyer les travaux sur les procédures de mis à l’essai des armes à impulsions (AI). Le présent rapport décrit le logiciel, sa structure et ses règles administratives; il répertorie également tous les éléments de données nécessaires pour appuyer entièrement les procédures de mise à l’essai publiées.

Importance: Le présent rapport est conçu pour appuyer l’Initiative stratégique sur les armes à impulsions (ISAI), qui prévoit la mise à l’essai d’un certain nombre d’AI et de faire l’analyse des données d’essai des anciens systèmes dans le but de mieux comprendre et de mieux interpréter le comportement de ces dispositifs.

Perspectives: ’ISAI prévoit utiliser le présent rapport pour appuyer l’objectif visant à élaborer et à mettre à jour de nouvelles procédures de mise à l’essai des AI et à conseiller les organismes canadiens chargés de l’application de la loi relativement à ces dispositifs.

Table of contents



List of figures

List of tables

This page intentionally left blank.

1 Introduction

1.1 Background

Several studies including the Braidwood Commission report[1], the Report of the Standing Committee on Public Safety and National Security of the Conducted Energy Weapon, the report of the Commission for Public Complaints against the RCMP and other provincial reports and coroners' recommendations have discussed the need for reliable uniform testing of Conducted Energy Weapons (CEWs) independent of the manufacturer. Recognizing this need, a team at Carleton University (lead by Professor A. Adler and Mr. D.P. Dawson) begun a research program to develop techniques and data analysis methods for data from CEWs. This research has led to several outputs: 1) development of CEW data acquisition and test systems [2], 2) Development of software to analyse CEW 3) organizing two Canadian CEW workshops, and, based on consensus at the workshop, 4) leading of a team to write a CEW Test Procedure [3].

This Test Procedure is designed to enable organizations across Canada to test CEWs in a reliable repeatable manner to determine whether they are operating within manufacturer's specifications. Test results so obtained will be usable in various ways. The test procedure contains a set of recommendations for measurement of the performance characteristics of conducted energy weapons. It represents the opinions of its authors, a group of subject matter experts who have been involved in research on or testing of CEWs, and is subject to the disclaimer (presented in section 0.0 of [3]).

To test CEW weapons data, an implementation of the analysis software has been created by Carleton University. This software may be used in the analysis of the stored data. It is available under an open source license from Dr. Andy Adler, Systems and Computer Engineering, Carleton University.

This report documents the Carleton University Conducted Energy Weapons (CEW) Test analysis software that was written to support the work on the CEW Test Procedure [3]. This report is designed to support the Conducted Energy Weapons Strategic Initiative (CEWSI), a project implemented within the Canadian Police Research Centre and managed by Defence Research Development Canada (DRDC) through the Centre for Security Science (CSS). The CEWSI project will be testing a number of CEWs and doing analysis on legacy test data with the objective of better understanding and interpreting the performance of the devices, developing an updated CEW Test Procedure and providing advice to Canadian Law Enforcement. A team of Carleton University researchers, led by Professor A. Adler, have conducted testing on CEWs, and developed a software package to capture, store and analyze the data. Additionally, this team were lead authors of the current CEW Test Procedure [3] as well as other papers. In order to help prepare CEWSI for future steps, a contract for this work was left to Carleton University to document the software, its business rules and the data elements required to support the Test Procedure [3].

1.2 Scope

This document aims to document the Carleton University CEW analysis software, with a focus on:

1. Development of a logical data model and physical schema of the software data elements
2. Document algorithms and business rules incorporated into the software
3. Identify the additional data elements and business rules that will be required to fully implement the data requirements of CEW Test Procedure [3].

1.3 Conducted Energy Weapons and Electrical Testing

Conducted Energy Weapons (CEWs) are increasingly purchased by police in many countries as a less-lethal force option. The most widely used CEWs are the M26 and X26 models from TASER International (TI), which work by firing two small darts (electrodes) attached to wires into the subject. The weapon then sends a pulsatile electrical current into the subject, designed to be sufficiently rapid and with enough energy to cause muscular incapacitation [4]. This incapacitation is intended to give an arresting officer time to gain control over the subject.

Electrical Testing of a CEW is designed to measure the electrical output characteristics of the weapon into a standardized load, and to calculate parameters to describe the electrical output which are relevant to the safety and physiological effectiveness of the weapon. A schematic diagram of a CEW attached to a test configuration is shown in figure 1:

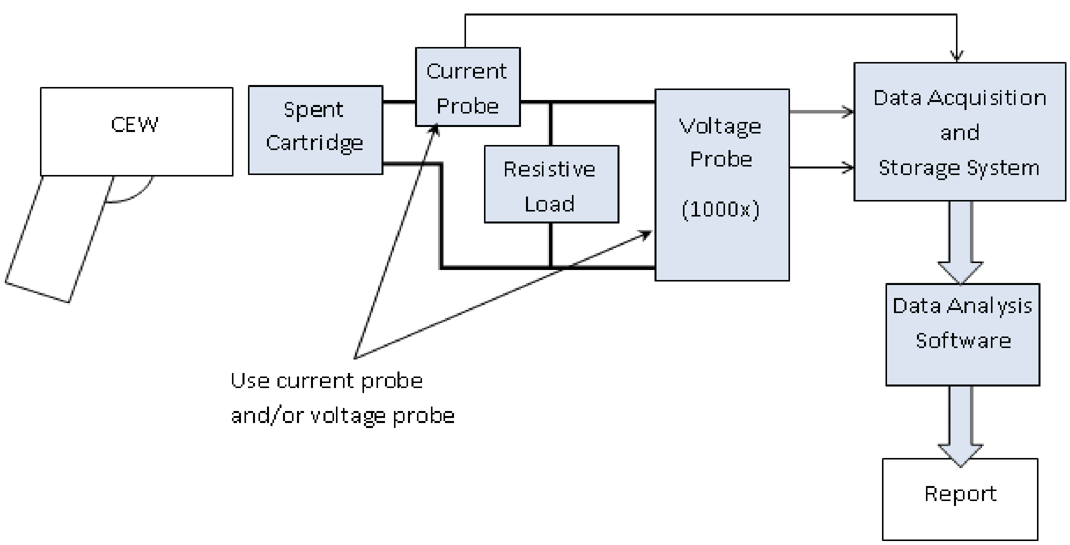


Figure 1 Schematic Diagram of a CEW attached to a test System (reproduced from [3])

The design specifications for the electronics hardware and analysis software in [3] are driven by a requirement to limit errors due to the data acquisition and processing to < 1%. However, it is assumed that current high voltage measurement probes (including voltage and current probes) can introduce errors up to 3%. Thus, the specifications of [3] were will result in only a small increase in errors due to the front end measurement probes.

The Carleton University team has been using a CEW measurement system based on a voltage probe (Tektronix 1000x probe 6015A). Since December 2010 Carleton University has been using a PicoScope dual channel 20MHz scope (model 4224) in combination with a Dell Vostro Laptop. Data is acquired at 10 MSamples/s with a precision of 12 bits. The PicoScope and laptop create a raw data file which is less than 500kB in size, by detecting and storing only periods in which there are pulses (ie energy) in the data. The system and file format are documented in [5]. From 2009 to 2010 the testing was done using a National Instruments PXI-5122 digitizer and PXI 1022 platform, sampling at 2 MSamples/s with 14 bit precision. Because the entire data stream was recorded, this produced data files in excess of 27MB making memory management a critical consideration.

1.4 Characteristics of the Taser X26 Waveform

This section provides details on the waveform, definitions and specifications of the parameters of interest for a TASER X26. The X26 pulse consists of an “Arc phase” and “Main phase” as shown in fig. 2. The pulses are delivered in a cycle consisting of approximately 95 pulses over 5 seconds, at the rate of approximately19 pulses per second. In a X26 pulse waveform, the information which electrically characterises the weapon is primarily derived from the Main phase, where most of the pulse energy resides. The Main phase delivers approximately 100 µC of charge, whereas the arc phase has only about 10 µC.

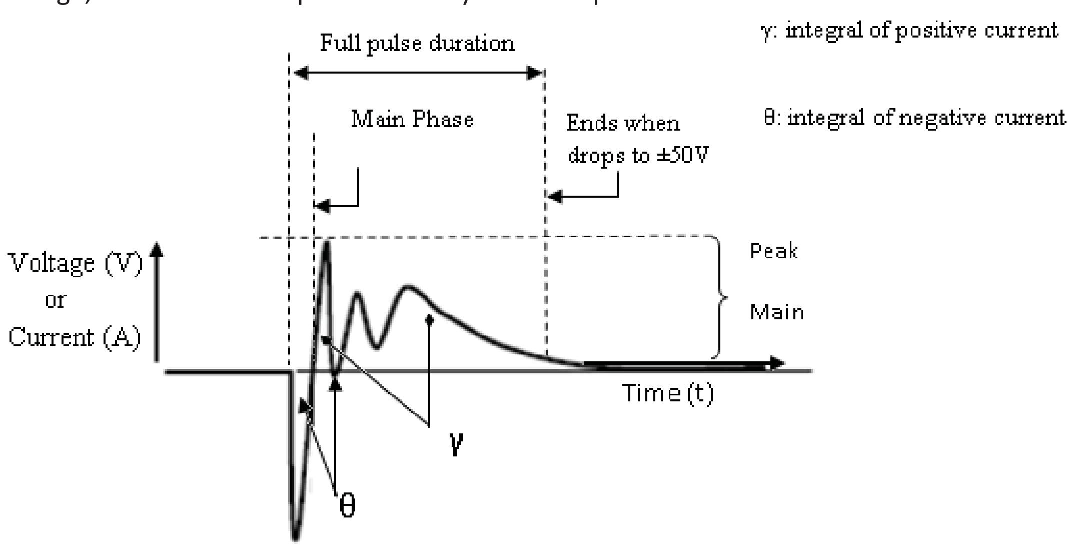


Figure 2 Taser X26 pulse waveform showing: (i) Peak voltage (or peak current) computed using the waveform Main phase, (ii) Net charge calculated as the absolute value of area under the current curve in the Main phase (γ), (iii) Full pulse duration starting at an initial crossing of ±50V

and finishing when pulse decreases to ±50V, (iv) Monophasic Charge calculated as the maximum of absolute values of γ and ϑ, (v) Total charge computed as the sum of the absolute values of γ and ϑ.

Table 1 shows the parameters of individual pulses which were identified in [3]. The list of pulse parameters was partially based on the definitions by Taser International [6]

Table 1 Definition of Parameters

Parameter	Definition
Peak voltage/current using the Main phase:	The peak voltage or peak current of an X26 pulse is computed using the Main phase of a voltage or current curve.
Net charge using the Main phase:	The net charge is computed using the Main phase of an X26 pulse. The Net charge is calculated as the area under the Main phase section starting at the first crossing of the Main phase above $0V$ ($0mA$) and finishing when pulse decreases to $83mA = 50V/600\Omega$.
Pulse duration using full pulse:	Pulse duration of an X26 pulse starting at an initial crossing of $\pm 50V$ and finishing when pulse decreases to +$50V$.
Pulse repetition rate:	The pulse repetition rate is computed by 1) counting the number of pulses in a full cycle, 2) measuring the cycle time from the start of the first pulse to the start of the last pulse and 3) the pulse repetition rate is one less than the total number of pulses, divided by the cycle time.
Monophasic Charge:	The Monophasic Charge is calculated as the maximum of the absolute values of γ and θ, where γ is the integral of the positive current in the X26 pulse waveform and θ is the integral of the negative current in the X26 waveform, respectively
Total Charge:	The total charge is calculated as the sum of the absolute values of γ and θ.

2 Carleton System

2.1 Test Environment at Carleton University

The Carleton University CEW test system was originally designed in 2009 and has seen incremental improvements over the past two years. Its design was based on the following guidelines:

1. Testing equipment is to be as mobile and portable as possible to allow testing to take place in front of the client and to minimize out-of-service time for the CEW.

2. At the same time, the test environment should preserve the integrity of measurement and analysis by adhering to sound engineering practices. This means that the components of the test system must tolerate movement in secure Pelican-type cases. The equipment and system as a whole must be configured and calibrated in the field as easily as it would in a static laboratory environment. Other variables such as temperature and humidity were measured and noted in each separate environment.

The test environment for *in-situ* testing is always indoors in a heated and/or air-conditioned secure space. Temperatures are generally 20C ± 2C, and humidity is generally between 20% and 40%. All test spaces are controlled areas where there is no public access and "technical tourism" is held to a minimum, and thus physical distractions are minimized.

Weapons are controlled by police officers and handed to the Testing Engineer by the officer for insertion in the mount. The CEW's are returned to their cases or storage mounts when not being physically examined or electrically tested. There are no special requirements for ventilation or air handling in the test environment because there are no harmful odours or particulate emissions for the firing of the weapon. There is no projectile from firing and there is no noise from firing into a calibrated load so there are no special requirements with respect to position of firing officer or safety equipment such as goggles or ear protection. The software and hardware used in the testing environment are controlled by the test engineer. There is no unauthorized use or modification of either hardware or software outside the purpose of testing CEW's for which it was intended.

2.2 Carleton University CEW analysis software

CEW testing at Carleton University is performed in two logically separate steps, as outlined in fig 3. First data are acquired using an acquisition hardware system. Data are digitized and stored to a file (the file format is documented in the next section, 2.3). Next, the data are analysed using the analysis software documented here. CEW pulse data from the raw data file are loaded, and pulse parameters calculated.

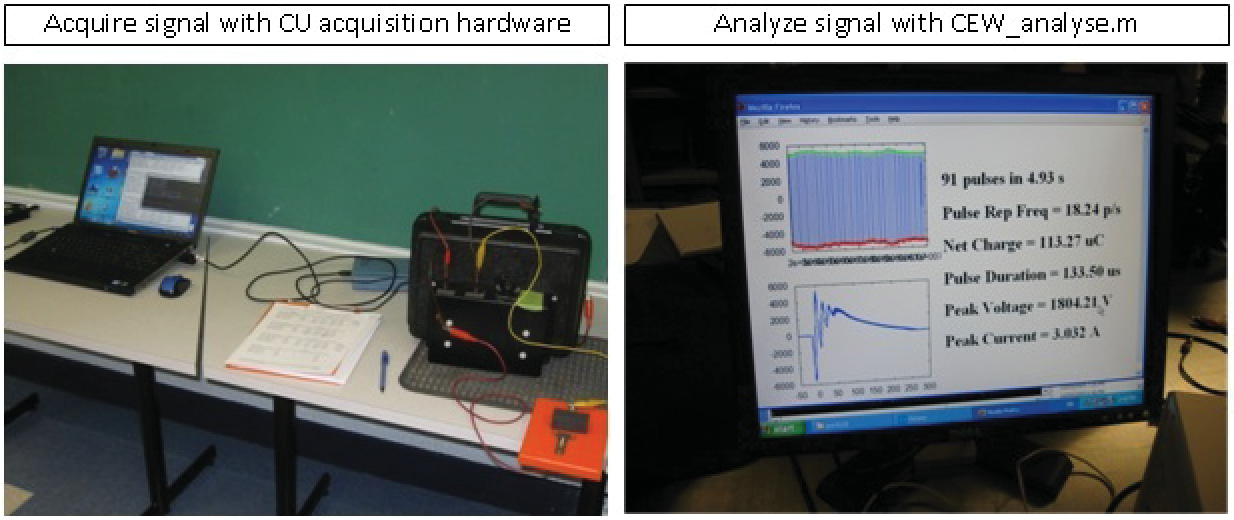


Figure 3 Schematic diagram of workflow of CEW testing at Carleton. Step 1: the CEW is connected to the electrical test system and data from a weapon firing is captured and stored to a raw data file. Step 2: the raw data file is loaded by the CEW_analyse *software and parameters are calculated.*

2.3 Carleton University CEW output file format

Data acquired by the CEW data capture hardware are written to a file with a format documented in [2] and in Table 2. This file format is designed to save storage space by storing only pulse data during the time that the weapon is active. Between pulses, when the output voltage is zero, the signal value is not stored.

Table 2 Raw Data file format

	Data Contained	**Data Type**	**Valid Data**
Header data	File Format	String	CU-CEWdata VER 1.00
	Date and time of Data collection	String	2010/10/26:14h34
	Serial Number	String	X00181701
	Description	String	A string of length < 65535 bytes. (length, string)
	Owner's name	String	A string of length < 65535 bytes. (length, string)
	Comments	String	A string of length < 65535 bytes. (length, string)

	Gain	Float (4 bytes)	Positive real values
	Offset (V)	Float (4 bytes)	Positive real values
	Resistance (Ω)	Float (4 bytes)	Positive real values
	Sampling Rate (Hz)	Float (4 bytes)	Positive real values
	Number of Pulses	Integer 32-bit	0 to $2^{31} - 1$
	Pulse length	Integer 32-bit	0 to $2^{31} - 1$
Interpulse Time (From the beginning of a pulse to the beginning of the next pulse)	t_1 (s)	Integer 16-bit	0to $2^{15} - 1$
	t_2 (s)	Integer 16-bit	0 to $2^{15} - 1$
	. .	Integer 16-bit	0to $2^{15} - 1$
	t_{n-1} (s)	Integer 16-bit	0 to $2^{15} - 1$
Interpulse Time (From the end of a pulse to the end of the next pulse)	t_1 (s)	Integer 16-bit	0 to $2^{15} - 1$
	t_2 (s)	Integer 16-bit	0 to $2^{15} - 1$
	. .	Integer 16-bit	0 to $2^{15} - 1$
	t_{n-1} (s)	Integer 16-bit	0 to $2^{15} - 1$
Sampling Data	$S_{1,1}$ (V)	Integer 16-bit	-2^{15} to $2^{15} - 1$
	. .	Integer 16-bit	-2^{15} to $2^{15} - 1$
	$S_{\text{Pulse length, 1}}$ (V)	Integer 16-bit	-2^{15} to $2^{15} - 1$
	$S_{1,2}$ (V)	Integer 16-bit	-2^{15} to $2^{15} - 1$
	. .	Integer 16-bit	-2^{15} to $2^{15} - 1$
	$S_{\text{Pulse length, 2}}$ (V)	Integer 16-bit	-2^{15} to $2^{15} - 1$
	$S_{1,n}$ (V)	Integer 16-bit	-2^{15} to $2^{15} - 1$
	. .	Integer 16-bit	-2^{15} to $2^{15} - 1$
	$S_{\text{Pulse length, n}}$ (V)	Integer 16-bit	-2^{15} to $2^{15} - 1$

2.4 Carleton University CEW analysis software

The software to perform the analysis of CEW waveforms is called CEW_analyse.m. It is designed to be run under Matlab or GNU Octave interpreters, and is used to analyse signals from CEW's acquired on the PicoScope. It comprises 382 lines and 13 functions. Table 3 sets out the names of the functions and what each function does, as well as some ancillary information such as line numbers, variables passed in, other functions called by the function being described. The MatLab code can be seen in annex A.

Table 3 CEW_analyse.m function descriptions

	Name of function	Lines	What it does	Calls other functions
1	CEW_analyse	12-65	For each data file entered the data is read and placed in to data and ss, finds the params for each data point plots the voltage at each point if requested and then call print_outputs	print_outputs read_data calibrate_data retrigger_data find_pts find_patarms print_output
2	print_outputs	70-173	Outputs the first two tables on page 13 of the test procedure, for each shot calculates interpulse length, total firing length, interpulse distance, median PRF, PRF from last second, set flag if PRF <16.5 prints a string header of table. Prints the third table on page 13 of the test procedure.	max_min_avg_avg8 outlines prt_log
3	outlines	177 – 193	Define column widths and string headers for tabular output in HTML (a.k.a page 13 output)	
4	find_pts	197-220	Defines 3 points on the waveform: start, zcross, endpoint	my_medfilt
5	find_patarms	225-242	Defines limits of arc phase and main phase and produces dT. Uses Vpk to calculate Ipk, PD, Qnet, Qmono, Qtotal	
6	system_settings	244-262	System settings for Picoscope	
7	getstr	265-267	Reads name from the files and gets a string of that length	
8	read_data	271-305	Reads text data from data file. Reads 9 system settings variables from data file. If no pulses detected print warning. If max_data exceeds clip limits, print warning. Print string identifying which data is being accessed	get_str prt_log
9	calibrate_data	311-321	Calibrates the data to the gains associated with the equipment being used and calculates the voltage from the data. Determines the time at each sampling point.	
10	retrigger_data	326-344	Reduces the length of data from 250 µs to 200 µs	
11	prt_log	347-354	Creates an output file and prints to it	
12	my_medfilt	358-365	Applies a median filter, each location the data is replaced with the median	

			value of the data points next to it.	
13	max-min-avg-avg8	370-382	Finds absolute maxima, absolute minima, mean and average of last 8 pulses. Sets flag if last 8 pulses exceed max value or are less than min value.	

2.5 Physical schema of the software data elements

Table 4 describes the variables used by each function in CEW_analyse.m. The description of each variable is given in Annex B.

Table 4 Physical schema overview

Function	Variables passed in	Variables at Output	Local Variables
CEW_analyse	f_name,options		ss, data, V, t, pts, params
print_outputs	params,ss		interpulse, PRF, mmaa, flag
outlines		o	
find_pts	Vi,ss	pts	start, zcross, endpt
find_patarms	Vi, t, pts, ss	pp	arc_phase, main_phase, dT
system_settings		ss	
getstr	fid	str	len
read_data	f_name ,ss	data, ss	fid
calibrate_data	data_input, ss	data, V, t	
retrigger_data	V, t, ss	Vo, t	thresh, Vi,
prt_log	str, varargin		txt, resultsfile
my_medfilt	s, n	s	ls, idx
max-min-avg-avg8	data, minval, maxval	m_m_a_a8,flag	

2.6 Logical Data Model

This section describes the logical data model of the software and its data elements of the CEW_analysis.m software. Figure 4 gives the underlying logical data model with which we were working for the design of the software.

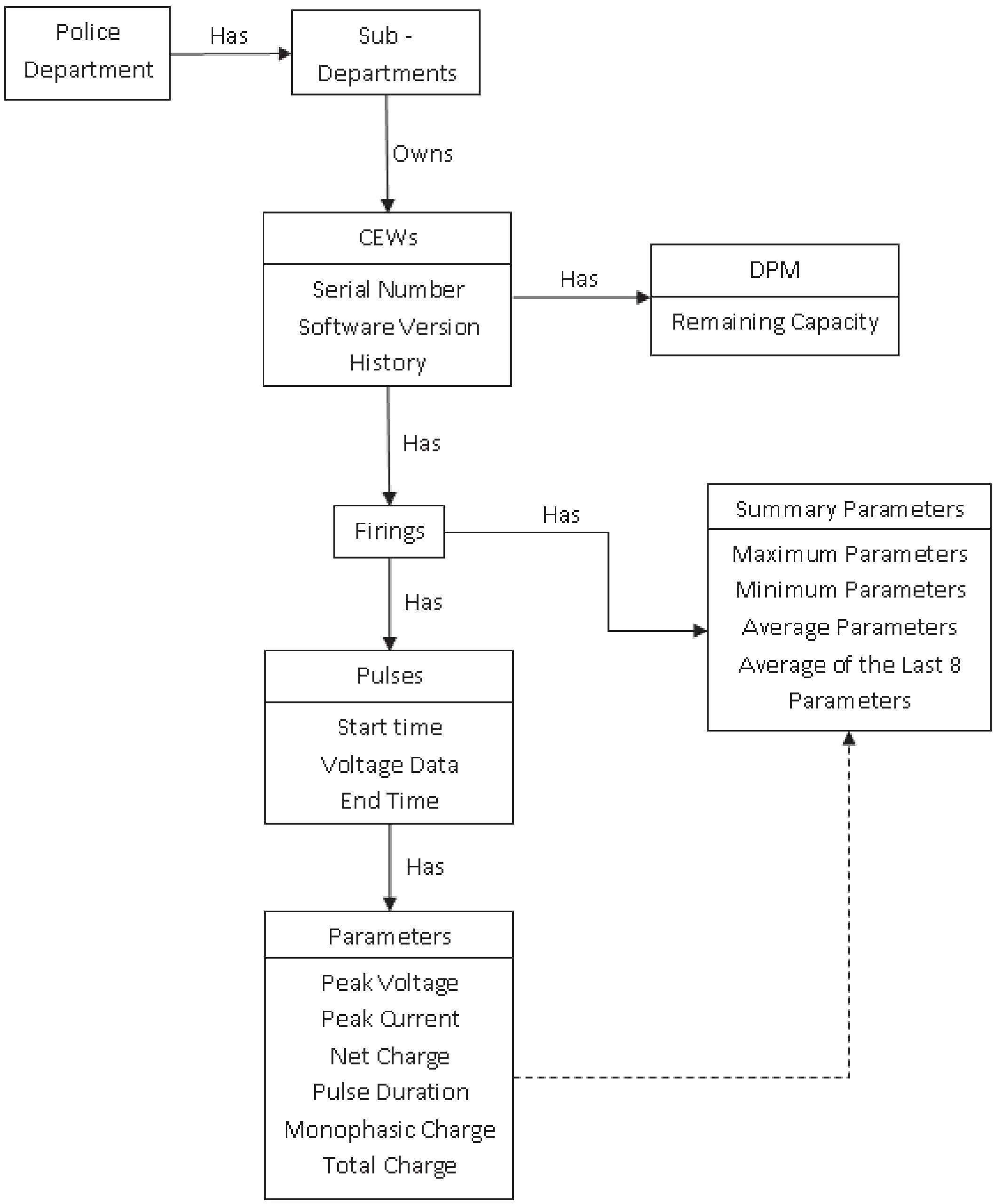


Figure 4 Schematic representation of the Logical Data Model.

2.7 Business rules incorporated into the software

2.7.1 Business Rules

The business rules used by Carleton in the analysis of a CEW are as follows:

- Equipment is setup and tested at the facility in which testing is to be conducted.
- The weapon owners are briefed about the test procedure to be followed.
- The Carleton Pico-scope acquisition software is executed on the PC.
- The Description and Owner of the CEWs being tested is entered
- The individual testing of each individual CEW begins and is repeated for each weapon
  - The serial number and comments for the CEW being tested is entered
  - Comments include any visual observations or special equipment such as handgrip camera
  - The program waits for the firing to begin
  - The trigger is pulled
  - The recording begins when the voltage crosses 100mV; this voltage is after a 1000:1 voltage reducing probe
  - System records each pulse from the weapon, starting at 50µs before this voltage crossing the trigger (using the pre-trigger feature of the hardware)and 200µs after the voltage trigger.
  - A trigger level of crossing (either +50V or –50V) is used for all pulses
  - Then the time between each pulse is measured
  - Recording of pulses stops if either of the following conditions are met: a) 120 pulses are recorded, or b) the testing user hits a key. In almost all cases, recording is stopped by the user.
  - The program then saves data to a raw data file in the format outlined in section 2.3
- The operator ensures that the raw data file and analysis code are placed in the same folder.
- The analysis code is then called in Matlab using the following call; CEW_analyse({file name one; file name two; file name three}, options), File names are with extensions and in single quotations, e.g. 'bin16out-(PoliceDepartment,X00-123456,97,21)-01276110295cvt.dat'
- For each pulse in the data stream, the parameters are calculated, as defined by. *Data points* refers to individual sampled voltage values (after correcting for the gain and offset):
  - *Peak voltage*: determined by the maximum data point on the data points in the Main phase (note: this is not the maximum absolute value).
  - *Peak current*: determined by the *peak voltage* divided by the load resistance.
  - *Pulse duration* is the time from the *pulse start point* to the *pulse end point.* It is

calculated by subtracting (*pulse end point*) – (*pulse start point*), in which the points are calculated as:

        - Pulse start point: the time at which the signal crosses from V>– 50 V to V<–50 V.

        - Pulse end point: the *last* time at which the *filtered* signal crosses from V>50V to V<50V. Filtering is a median filter in which each point is the median of the samples within a width of 2 us in each direction. The *last* crossing is identified by, first, identifying all crossings of the 50V threshold by the filtered signal in each pulse, and, then, taking the last one.

    - *Main Phase*: The main phase is the time from the *main phase start* to the *pulse end point* (defined above).

        - The *main phase start* is the first time the signal crosses from V<0V to V>0V after the *pulse start point*

    - *Net Charge* is calculated by summing the data points in the *main phase* multiplied by the sampling period and divided by the resistance (note: this is the sum, not the sum of absolute values)

    - *Monophasic charge* is calculated by the maximum of A and B, where: A is the sum of the absolute value of all data points with V>0 during the *pulse duration* multiplied by the sampling period, and divided by the resistance; and B is the sum of the absolute value of all data points with V<0 during the *pulse duration* multiplied by the sampling period, and divided by the resistance

    - *Total charge* is calculated by summing the absolute value of data points during the *pulse duration* multiplied by the sampling period, and divided by the resistance.

    - *Interpulse time* for pulse N is calculated as the time from *pulse N start point* to *pulse N+1 start point.*

- Based on the parameters calculated for each pulse, summary parameters are calculated for the entire pulse train for the weapon firing.

    - For parameters *Peak voltage, Peak current, Net Charge,* and *Pulse Duration*, the following summary parameters are calculated

        - *Maximum*, the maximum of all pulse values

        - *Minimum*, the minimum of all pulse values

        - *Average,*, the average (mean) of all pulse values

        - *Average–TI*, the average (mean) of the last 8 pulse values

    - For parameters *Total Charge,* and *Monophasic Charge*, the following summary parameters are calculated

        - *Maximum*, the maximum of all pulse values

        - *Minimum*, the minimum of all pulse values
        - *Average*, the average (mean) of all pulse values
    - Based on the parameter *interpulse time,* the a *pulse rate* is calculated for each pulse as 1/(*interpulse time).* Using this value, the following summary parameters are calculated
        - *Maximum*, the maximum of all pulse values
        - *Minimum*, the minimum of all pulse values
        - *Average,* the average (mean) of all pulse values
        - *Burst Length,* the sum of all interpulse time values
        - *Average–TI*, the (Number of pulses – 1) / (*burst length*)
- Summary parameters are compared to the specified values in the appendix A (TASER M26) and appendix B (TASER X26) of [1]. Values outside of the specified range are identified (with the string '****')
- Individual pulse and summary parameters are saved to a file
- Summary parameters are displayed to the user and saved to a file

2.8 Additional data elements and business rules that will be required to fully implement the data requirements of CEW Test procedure

The current version of CEW_analyse.m and test apparatus used by Carleton in the testing of CEW requires some additional data elements to conform the Carleton test procedure, this can be seen in Table 5.

5. Requirements to conform to the Carleton test Procedure

Data Element	**Comments**
Temperature	It is currently being recorded by hand on to paper. It should be in the raw data file with the other system settings
Humidity	It is currently being recorded by hand on to paper. It should be in the raw data file with the other system settings
Remaining battery capacity	It is not currently being recorded; it has been seen to have no effect on how the weapon performs. It should be recorded and placed into the raw data file with the other system settings so as to conform to the

	Carleton test procedure.
Software Version	It is not currently being recorded. It should be recorded to conform to the Carleton test procedure. In practice, most police departments are buying the latest DPM which contains the most recent software and cycling this DPM through the inventory of CEW's.
Owner of the CEW	The owner of the CEW is currently in the raw data file but is not being consistently recorded while testing.

Summary and Conclusions

This report describes the test analysis software developed at Carleton University (CEW) that was written to support the work on the Conducted Energy Weapons (CEW) Test Procedure, [3]. The objectives of the report is to document the software, its structure, business rules, and to identify any data elements required to fully support the published test procedure. Within the report, each of these objectives is systematically described.

This report is designed to support the Conducted Energy Weapons Strategic Initiative (CEWSI), which plans to test a number of CEWs and do analysis on legacy test data with the objective of better understanding and interpreting the performance of the devices, with a goal of developing an updated CEW Test Procedure and providing advice to Canadian Law Enforcement on these devices.

References

[1] Thomas R. Braidwood, “Braidwood Inquiry”, Commissions of Inquiry, Vancouver, BC, May 20, 2010. Online: http://www.braidwoodinquiry.ca/report/P2Report.php

[2] Peyman Rahmati, David P Dawson, Andy Adler, “Towards a Portable, Memory-Efficient Test System for Conducted Energy Weapons”, Canadian Conference on Electrical and Computer Engineering (CCECE 2011), Niagara Fall, Canada, May 8-11, 2011

[3] Andy Adler, Dave Dawson, Ron Evans, Laurin Garland, Mark Miller, Ian Sinclair, “Test Procedure for Conducted Energy Weapons”, Version 1.1: 31 July 2010. Online: http://curve.carleton.ca/papers/2010/CEW-Test-Procedure-2010-ver1.1.pdf

[4] Panescu D, Stratbucker RA. “Current Flow in the Human Body.” in Kroll MW, Ho JD (Eds) *Taser Conducted Electrical Weapons: Physiology, Pathology and the Law*, Springer, 2009.

[5] David P Dawson, Andy Adler, “Towards a Test Standard for Conducted Energy Weapons”,

[6] Taser International, *TASER X26E Series Electronic Control Device Specification Version 2.0,* Feb 6 2009. Online: ecdlaw.info

[7] European Working Group on Non Lethal Weapons − 6th European Symposium Ettlingen, Germany, May 16-18, 2011

[8] David Dawson, Yasheng Maimaitijiang, Andy Adler, “Development of a Performance Calibration System for X-26 Tasers”, Int. Workshop Medical Meas. and Appl., Ottawa Apr 30-May 1, 2010

Annex A MATLAB code for CEW_analyse.m

```
%%This code is designed to analyse CEW X26 weapons according to the test
%%procedure found online at:
%%http://curve.carleton.ca/papers/2010/CEW-Test-Procedure-2010-ver1.1.pdf
%%(c) A. ADLER and O. MARSH, 2009-2011
%%This code may be copied under the GPL licence version 2 or 3 found at:
%%http://www.gnu.org/licenses/gpl.html

%%Main function of CEW_analyse reads data from the file in f_name and places
%%it in data and ss, checks for the presence of energy in the file,
%%finds V at all sample points, plots all voltage values and repeats for the
%%files entered in to the function then calls print_outputs
function CEW_analyse(f_name, options);

ss = system_settings;

if nargin>1;
  ss.plotting = options.plotting;
  ss.debug_plot = options.debug_plot;
  ss.use_filter = options.use_filter;
  ss.open = options.open;
else
  ss.plotting = 1;
  ss.debug_plot = 0;
  ss.use_filter = 0;
  ss.open = 1;
end

if ss.plotting; clf; end

n_shot = size(f_name);
n_shot = n_shot(1,1);
if n_shot>3;
  fprintf('Too many files entered only first three will be used\n');
  n_shot = 3

elseif n_shot == 2;
  ss = [ss, ss];
elseif n_shot == 3;
  ss = [ss, ss, ss];
end

for j = 1:n_shot;

  if ss(j).plotting; figure(j); end

    [data, ss(j)] = read_data( char(f_name(j)), ss(j));

    [data,V,t] = calibrate_data( data, ss(j));
    [V,t] = retrigger_data(V,t ,ss(j));

    for i=1:ss(j).pulses
```

```
      Vi = V(:,i);
      if all(Vi==0); break; end
      pts = find_pts( Vi, ss(j) );
      params(j, i) = find_patarms( Vi, t, pts, ss(j) );
      if ss(j).plotting
        ii= [pts.start,pts.zcross, pts.endpt];
        plot(t,Vi,t(ii), Vi(ii),'*');
        if i==1; hold on;end
      end
  end
end
print_outputs( params,ss)
fprintf('Calculations complete\n');
if ss(1).open; open('summary_results.html');end


%%print_outputs prints the results of the calculations in html format
%%which follows the format of page 13 in CEW test procedure
function print_outputs( params, ss)
table12 = strcat('<table width = "800" border="2" rules = "all" bordercolor =" black"><tr>',...
'<td width = "500"><b> Conductive Energy Weapon Test Report <b/> <td width = "300">',...
'Date: %s <tr> <td > Weapon: <td > Serial Number: %s<tr> <td > Police Service: %s',...
'<td > Police Officer: <tr><table/><br>',...
'<table width = "800" border="2" rules = "all" bordercolor ="black" ><tr>',...
'<td width = "200" >Visual Inspection<td width = "600"> Case Battery Electrodes',...
'<tr><td> Data Download Performed <td><tr><td> Comments <td>%s<tr>',...
'<td> Software Version <td><tr><td> Battery Charge <td><tr>',...
'<td> Battery Model and Serial<td><tr><td> Temperature <td><tr><td> Humidity<td>',...
'<tr><td> Atmospheric Pressure <td><table/><br>');

prt_log(table12, ss(1).meastime,ss(1).serial,ss(1).owner, ss(1).comment);
clear table12


n_shot = size (params);
n_shot = n_shot(1);
interpulse = zeros(size (params));
for i=1:n_shot
  interpulse(i,1:length(ss(i).interpse_e(2:end-1))) = ss(i).interpse_e(2:end-1);
  firinglength_2= sum(interpulse(i, :));
  firinglength = firinglength_2*(ss(i).pulses)/(ss(i).pulses-2) + mean ([params(i,:).pulse_duration]);

  medPRF= 1/median(interpulse(i));
  PRF = (ss(i).pulses-1) / firinglength;

  if PRF<16.5; flag = '****';
```

```
  else flag = '';end

  prt_log('Shot %d: %d pulses in %4.2f s. Pulse Rep Freq = %4.2f p/s
%
s<br><br>', ...
  i, ss(i).pulses, firinglength, PRF, flag);

end

oo = outlines;

%creating Third output table
table3 = strcat('<table width = "800" border="2" rules = "all"
bordercolor ="
black"><tr>',...
'<td width = "150"><th width = "150" colspan="%d"> Max <th width =
"150" colspan="%d">',...
'Min<th width = "150" colspan="%d"> Avg<th width = "150" colspan="%
d"> Avg - TI',...
'<td width 50> SPEC <tr><td > Firing No');

table3 = sprintf(table3, n_shot, n_shot, n_shot, n_shot);

for i=1:4
  for j = 1:n_shot
    table3 = sprintf(strcat(table3, '<td align = "center">%d'),j);
  end
end

table3 = strcat(table3, '<td>');

prt_log(table3);

flag.pv = ''; flag.pc = ''; flag.nc = ''; flag.pd = ''; flag.pr ='';
flag.tc = ''; flag.mc = '';
for i = 1:n_shot
  [mmaa(i,:).pv,tflag] = max_min_avg_avg8(
[params(i,:).peak_voltage],
1400,2520 );
  flag.pv = strcat(flag.pv, tflag);
  [mmaa(i,:).pc,tflag] = max_min_avg_avg8(
[params(i,:).peak_current], 2.3,
4.2 );
  flag.pc = strcat(flag.pc, tflag);
  [mmaa(i,:).nc,tflag] = max_min_avg_avg8(
[params(i,:).net_charge]*1e6,80,
125 );
  flag.nc = strcat(flag.nc, tflag);
  [mmaa(i,:).pd,tflag] = max_min_avg_avg8(
[params(i,:).pulse_duration]
*1e6,105, 155 );
  flag.pd = strcat(flag.pd, tflag);
  [mmaa(i,:).pr ,tflag] = max_min_avg_avg8(
[1./nonzeros(interpulse(i,:))],
0, inf );
  flag.pr = strcat(flag.pr, tflag);
  [mmaa(i,:).tc,tflag] = max_min_avg_avg8(
[params(i,:).total_charge]*1e6,
0, inf );
  flag.tc = strcat(flag.tc, tflag);
```

```
  [mmaa(i,:).mc,tflag] = max_min_avg_avg8( [params(i,:).monophasic_charge] *1e6,0, 180 );
  flag.mc = strcat(flag.mc, tflag);
end

prt_log(oo.pv);
prt_log(oo.table, mmaa.pv);
prt_log(oo.flag, flag.pv);

prt_log(oo.pc);
prt_log(oo.table, mmaa.pc);
prt_log(oo.flag, flag.pc);

prt_log(oo.nc);
prt_log(oo.table, mmaa.nc);
prt_log(oo.flag, flag.nc);

prt_log(oo.pd);
prt_log(oo.table, mmaa.pd);
prt_log(oo.flag, flag.pd);

prt_log(oo.pr);
prt_log(oo.table, mmaa.pr);
prt_log(oo.flag, flag.pr);

prt_log(oo.tc);
prt_log(oo.table, mmaa.tc);
prt_log(oo.flag, flag.tc);

prt_log(oo.mc);
prt_log(oo.table, mmaa.mc);
prt_log(oo.flag, flag.mc);

prt_log('</table>');
return


%%outlines defines the formatting of the tabular HTML output
function o = outlines;
  o.table = '<td align = "center"> %.2f';
  o.flag = '<td align = "center"> %s';
  % peak voltage (main phase)
  o.pv='<tr><td > Peak Voltage (V)';
  % peak current (main phase)
  o.pc= '<tr><td >Peak Current (A) ';
  % net charge (main phase)
  o.nc= '<tr><td >Net Charge (&#181;C)';
  % pulse duration (full pulse),
  o.pd= '<tr><td >Pulse Duration (&#181;s)';
  % pulse repitition rate
  o.pr= '<tr><td >Pulse Rep Rate (Hz)';
  % Monophasic Charge
  o.mc= '<tr><td >Mono. Charge (&#181;C)';
  % Total Charge
  o.tc= '<tr><td >Total Charge (&#181;C)';

%%find_pts finds the location of three points on the waveform, the start
%%point, first zero crossing, and the end point
function pts = find_pts( Vi, ss );
```

```
  Vi = my_medfilt( Vi, 0);
  start = find(Vi<=-50);
  start = start(1);


  zcross= find(Vi>=0);
  zcross(zcross<start)= [];
  zcross = zcross(1);

  Vi = my_medfilt( Vi, 20);
  endpt = find(Vi < 50);
  delay = 30e-6*ss.sample_f; % 30 us
  endpt(endpt< (zcross + delay)) = [];
  if isempty(endpt)
    endpt = 190e-6*ss.sample_f-1
  else
    endpt= endpt(1);
  end


  pts.start = start-1;
  pts.zcross= zcross;
  pts.endpt = endpt;

%%find_patarms uses the data for each pulse to calculate peak voltage,
%%peak current, net charge, pulse duration, monophasic charge and total
%%charge
function pp = find_patarms( Vi, t, pts, ss );
  arc_phase = Vi(pts.start: (pts.zcross -1));
  main_phase = Vi(pts.zcross:pts.endpt);
  dT = mean(diff(t));

  % peak voltage (main phase)
  pp.peak_voltage = max(main_phase);
  % peak current (main phase)
  pp.peak_current = pp.peak_voltage / ss.resistance;
  % net charge (main phase)
  pp.net_charge = sum( main_phase ) * dT / ss.resistance;
  % pulse duration (full pulse),
  pp.pulse_duration = t(pts.endpt) - t(pts.start);
  % Monophasic Charge
  pp.monophasic_charge = sum( Vi .* (Vi>0) )*dT / ss.resistance;
  % Total Charge
  pp.total_charge = sum( abs( Vi ) )*dT / ss.resistance;

%%system_settings defines the default settings of the testing apparatus
function ss= system_settings
  ss.sample_f=10e6;
  ss.resistance = 595;
  ss.gain = 1/1000;
  ss.offset = 0; %actually -6mV
  ss.probegain=1000;
  ss.input_gain=1;
  ss.clip_limit = 2^(12-1) - 1;
  ss.fileformat = '';

  ss.meastime = '';
  ss.serial = '';
```

```
  ss.description= '';
  ss.owner = '';
  ss.comment = '';
  ss.pulses = 0;
  ss.pulse_len = 0;
  ss.interpse_s = 0;
  ss.interpse_e = 0;

%%getstr reads a length of a string for a file and then the string
function [str] = getstr(fid)
  len= fread(fid,1,'int16');
  str= char(fread(fid,[1,len],'char*1'));

%%read_data reads the data and system settings from the file and
checks
%%if there is data in the file and if signal is vertically clipped
function [data,ss] = read_data( f_name, ss);
  fid = fopen(f_name,'rb');
  ss.fileformat = char(fread(fid,[1,20],'char*1'));
  if ss.fileformat(end)~=0; error('format');end
  ss.meastime = getstr(fid);
  ss.serial = getstr(fid);
  ss.description= getstr(fid);
  ss.owner = getstr(fid);
  ss.comment = getstr(fid);
  ss.gain = fread(fid,1,'float');
  ss.offset = fread(fid,1,'float');
  ss.resistance = fread(fid,1,'float');
  ss.sample_f = fread(fid,1,'float');
  ss.pulses = fread(fid,1,'int32');
  ss.pulse_len = fread(fid,1,'int32');
  ss.interpse_s = fread(fid,ss.pulses,'float');
  ss.interpse_e = fread(fid,ss.pulses,'float');
  data = fread(fid,[ss.pulse_len,ss.pulses],'int16');

  fclose(fid);
  if ss.pulse_len ==500
    ss.clip_limit = 32764;
  end
%-------------------Analysing the stripped data------------------+
  max_data = max(data);
  min_data = min(data);
  if max(abs([max_data,min_data]))<1000
    prt_log('WARNING: No pulses detected in data file<br>');
    data= [];
    return
  elseif (max_data>=ss.clip_limit) |(min_data<-ss.clip_limit)
    prt_log('<br>WARNING: Signal is vertically clipped<br>');
  end

  prt_log('<br>Data loaded: file=%s<br><br>',f_name);


%%calibrate_data calibrates the data to the gains associated with
the
%%testing apparatus and calculates the voltage and time at each
%%sampling point
function [data,V,t] = calibrate_data( data_input, ss);
  if ss.use_filter
    [data, ip] = func_despike_phasespace3d( double(data_input), 1,2
);
```

```
  else
    data = double(data_input);
  end


  data= (ss.input_gain)*data;
  V = (data*ss.gain+ss.offset)*ss.probegain;
  t = (0:ss.pulse_len-1)/ss.sample_f;

%%retrigger_data takes the 250us's of sample points and reduces to 200us
%%of data sample points. It takes 10us before the pulse begins and 190us
%%after it begins
function [Vo,t] = retrigger_data(V, t, ss);
  thresh = 100;
  lowlim = int16(10e-6*ss.sample_f);
  VN = size(V,2);
  idx = -lowlim:190e-6*ss.sample_f-1;%spanning 200us
  Vo = zeros(length(idx),VN);

  for i=1:VN
    Vi = V(:,i);
    ff = find( abs(Vi) > thresh);
    if length(ff)>=1
      ff= ff(1);
      Vo(:,i) = Vi( ff+idx );
      lim = int16(11e-6*ss.sample_f);
      Vo(lowlim:lim,i) = linspace(Vo(lowlim,i),Vo(lim,i),lim-(lowlim-1));
    end
  end

  t= t(1:length(idx));

%%prt_log outputs the string called in to the function to output file
function prt_log(str, varargin );
  txt= sprintf(str,varargin{:});


  resultsfile = fopen('summary_results.html','at');
  fprintf(resultsfile, txt);
  fclose(resultsfile);

%%my_medfilt is a median filter to array s entered into the function
%%each data point is replaced with the median data point of the 2n+1
%%data points to either side of it
function s = my_medfilt( s, n);
  ls = length(s);
  idx= ones(2*n+1,1)*(1:ls) + (-n:n)'*ones(1,ls);
  idx(idx<1) = 1;
  idx(idx>ls) = ls;
  s = reshape(s(idx), [], ls);
  s = median(s,1);

%%max_min_avg_avg_avg8 takes data and determines the absolute minimum,
%%absolute maximum and mean of all pulses. Then determines mean of the
```

```
%%last eight pulses. It will throw a flag up if it is greater than the
%%maximum or minimum values.
function [m_m_a_a8,flag] = max_min_avg_avg8( data, minval, maxval )
  m_m_a_a8(1) = max(data);
  m_m_a_a8(2) = min(data);
  m_m_a_a8(3) = mean(data);
  m_m_a_a8(4) = mean(data(end-(0:7)));

  flag = '';
  if m_m_a_a8(4)>maxval;
    flag= '*';
  end
  if m_m_a_a8(4)<minval;
    flag= '#';
  end
```

Annex B Physical schema of the software data elements

Data Elements used by CEW_analyse:

Data Element	What it is	Type	Example
f_name	The File name of the file to be analysed	A array of strings	CEW-x00526878-(Thu_Dec_16_15.27.47_2010).dat
options	An Object contain the options for it plotting, debup_plot, use filter, and open results file	An object with 4 Boolean values contained in it.	options.plotting = 0 options.debug_plot = 0 options.use_filter = 0 options.open = 0
ss	The meta data for the testing	An object with 17 fields with variety of values stored	ss.meastime = '2010/10/26:14h34'; ss.serial = 'x00526878'; ss.description= 'ops'; ...
data	Raw data from the file	An array of int-16 values	assorted values ranging from -2^{15} to $2^{15} - 1$
V	Data calibrated to the associated gains	An array of int-16 values	assorted values ranging from -2^{15} to $2^{15} - 1$
t	Times at each point in the wave form	An array of floating point numbers	values ranging from 0s to 200us spaced evenly
pts	The points on the waveform at with the wave starts, crosses the zero point for the first time and the end point	An object containing 3 integer values.	Values range from 0 to 2000
params	An array containing the characteristics for each waveform in the pulse train	An array of objects with each object containing 6 float fields	params(1).peak_voltage = 1850 params(1).peak_current = 3.1 params(1).net_charge = 105.3e-6 params(1).pulse_duration = 125.2e-6 params(1).monophasic_charge = 110.4e-6

			params(1).total_charge = 120.1e-6

Data elements used by print_outputs: params and ss are equivalent to the fields in CEW_analyse.

Data Element	What it is	Type	Example
interpulse	Values of the time between pulses	An array of float values	values ranging from 0.04s to 0.2s
PRF	the Pulse repetition frequency of the firing	An float value	accepted values range from 16.5-20pps
mmaa	values of the mean, median, average and average of the last 8 for each parameter	An array of an object containing 7 fields each being an array of floats	mmaa.pv = 2000.3 1600.4 1850.5 1840.7 ...
Flag	flag for if any parameter is out of tolerance	An object with 7 fields with are strings	flag.pv = ‘****’ flag.nc = ‘####’ ...

Data elements used by outlines:

Data Element	What it is	Type	Example
O	The definition for the html output for each parameter	An object containing 9 fields each being a string.	o.table = '<td align = "center"> %.2f' o.pv = ‘<tr><td>Peak Voltage (V)’ ...

Data elements used by find_pts: ss and pts are equivalent to the fields in CEW_analyse.

Data Element	What it is	Type	Example
Vi	The voltage data for the pulse	An array of integers	Assorted values ranging from -2^{15} to $2^{15}-1$
start	The index in the array at which the pulse starts	An integer	Normal values range from 0-1000
zcross	The index in the array at which the pulse crosses zero for the first time in the waveform	An integer	Normal values range from 0-1000

endpt	The index in the array at with the pulse ends	An integer	Normal values range from 1000-2000

Data elements used by find_patarms: ss, pts and t are equivalent to the fields in CEW_analyse. Vi is equivalent to the field in find_pts.

Data Element	What it is	Type	Example
pp	The parameters (pv, pc, nc, pd, mc and tc) of the waveform	An object with 6 float fields.	pp.peak_voltage = 1850 pp.peak_current = 3.1 pp.net_charge = 105.3e-6 pp.pulse_duration = 125.2e-6 pp.monophasic_charge = 110.4e-6 pp.total_charge = 120.1e-6
arc_phase	the voltage values for the data points in the arc phase of the pulse	An array of int-16	Assorted values ranging from -2^{15} to $2^{15}-1$
main_phase	the voltage values for the data points the main phase of the pulse	An array of integers	Assorted values ranging from -2^{15} to $2^{15}-1$
dT	The time increments for between the data points	A float value	0.1us

Data elements used by system_settings: ss is equivalent to the field in CEW_analyse

Data elements used by getstr:

Data Element	What it is	Type	Example
fid	File id of the file being read from	File id	N/a
str	The string taken from the file	A string	'x00526878'
len	Length of the string being read from the file	A int-16	9

Data elements used by read_data: f_name, ssand data are equivalent the field in CEW_analyse. fid is equivalent the field in getstr.

Data elements used by calibrate_data: ss, data, V and t are equivalent the field in CEW_analyse.

Data Element	What it is	Type	Example
data_input	The raw data which needs to be calibrated	An array of int-16 values	assorted values ranging from -2^{15} to $2^{15} - 1$

Data elements used by retrigger_data: V, t and ss equivalent the field in CEW_analyse. Vi equivalent to the field in find_pts.

Data Element	What it is	Type	Example
Vo	The data points after the data has been retriggered	An array of int-16	Assorted values ranging from -2^{15} to $2^{15} - 1$
thresh	The threshold voltage to determine when the pulse begins	An int-16 value	100

Data elements used by prt_log:

Data Element	What it is	Type	Example
str	The string to be outputted	A string	'<td align = "center">%d'
varargin	Any values required to be outputted	Any number of values of any type	1
txt	the output string with the values added	A string	'<td align = "center">1'
resultsfile	the file id of the output file	file id	N/A

Data elements used by my_medfilt:

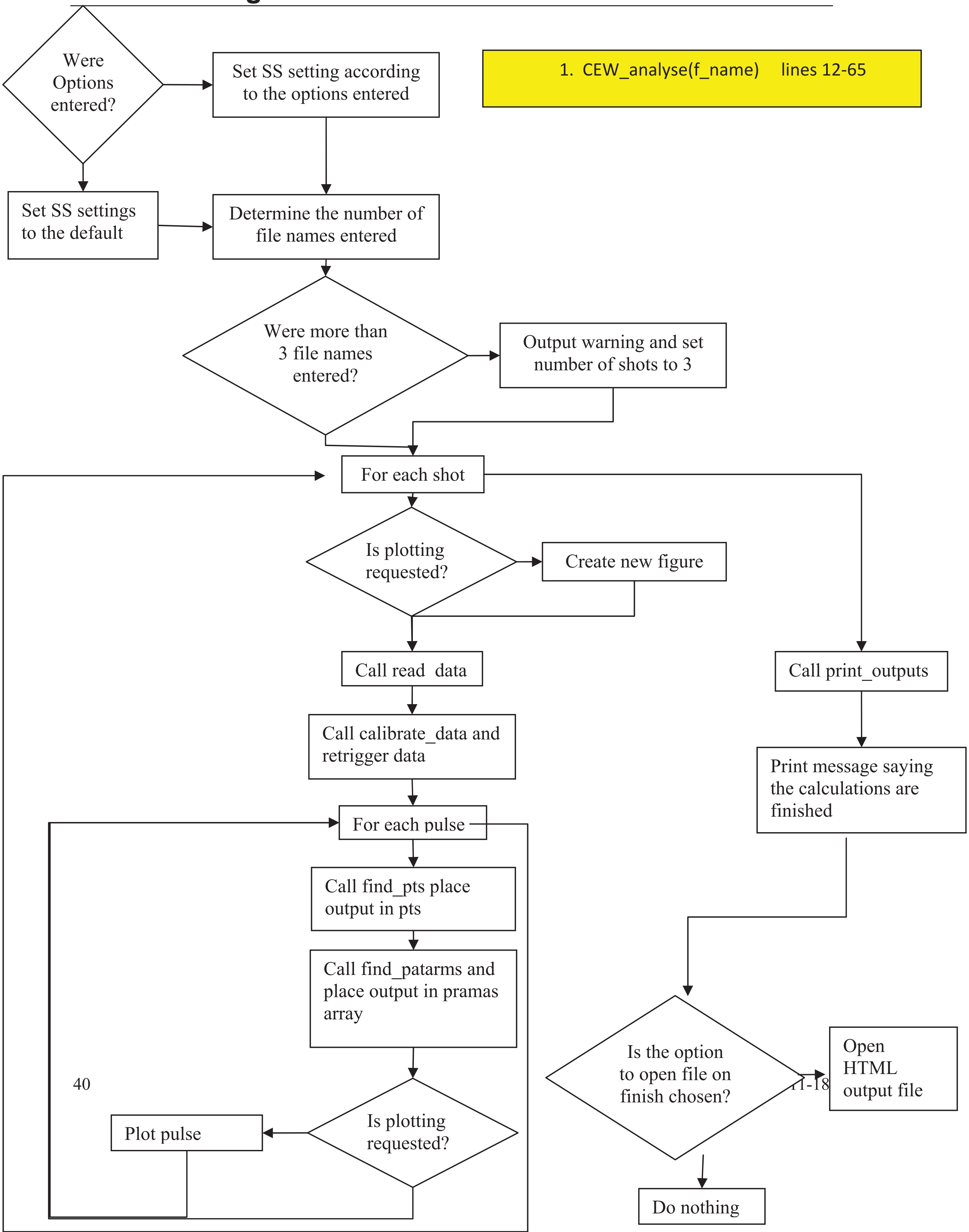
Data Element	What it is	Type	Example
s	The data points which	An array of int-16	Assorted values ranging

	need to be filter		from -2^{15} to $2^{15}-1$
n	Number of points on each side to take the median from	An int-16	The normal value is 40
ls	Number of data points in s	An int-16	Normally 2000

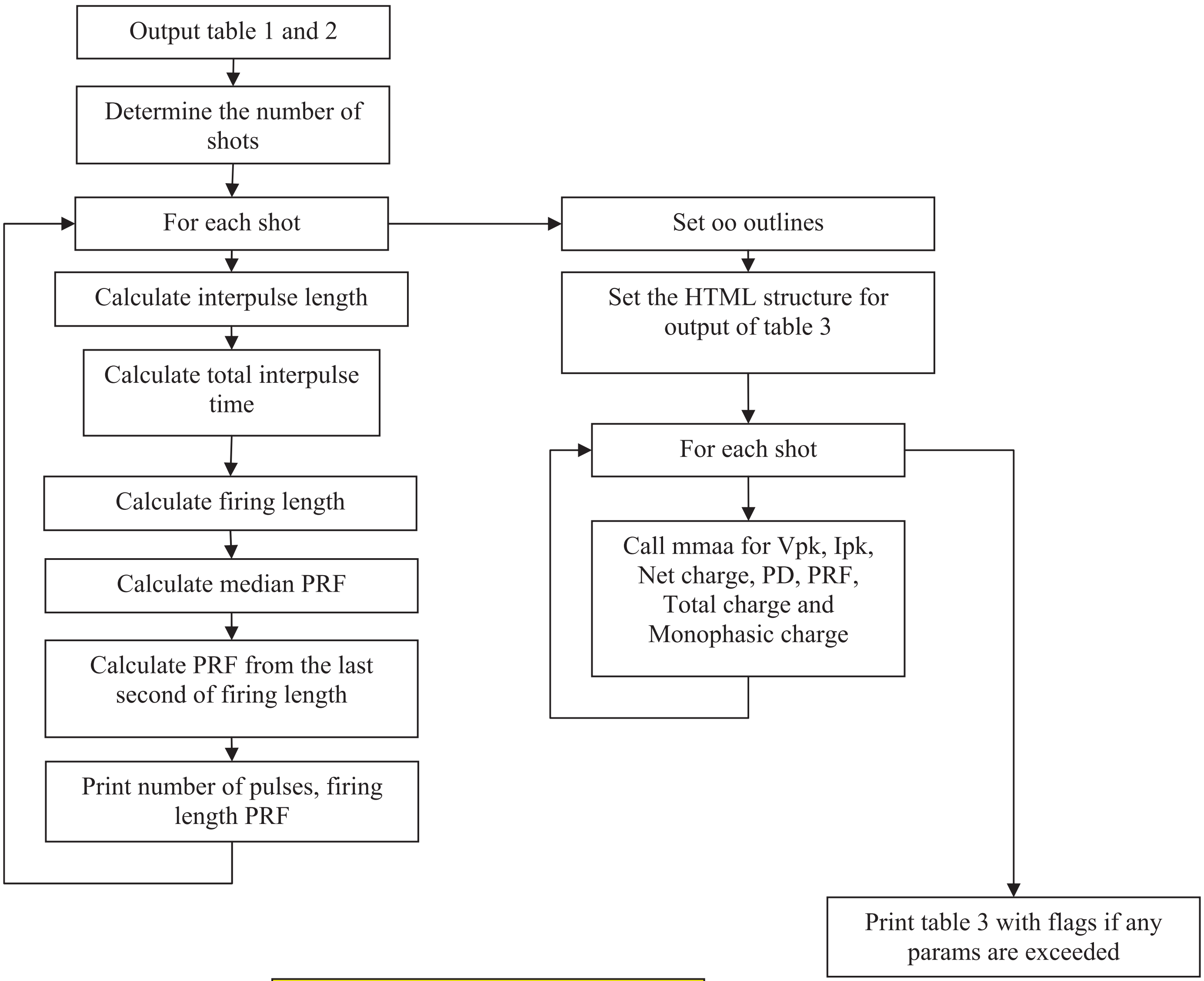
Data elements used by max-min-avg-avg8:

Data Element	What it is	Type	Example
data	The parameter values for each pulse	An array of floats	Assorted values
minval	The lower spec limit	A float	For Peak Voltage it is 1400
maxval	The upper spec limit	A float	For Peak Voltage it is 2520
m_m_a_a8	The mean, median, average and average-8 values	An array of floats	Assorted values
flag	Flag if any parameters are out of spec	A string	'****'

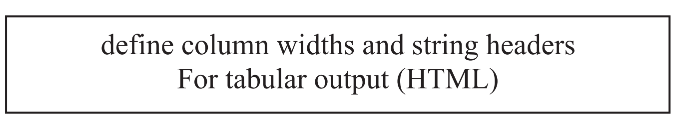
Annex C   Algorithms for Each Function

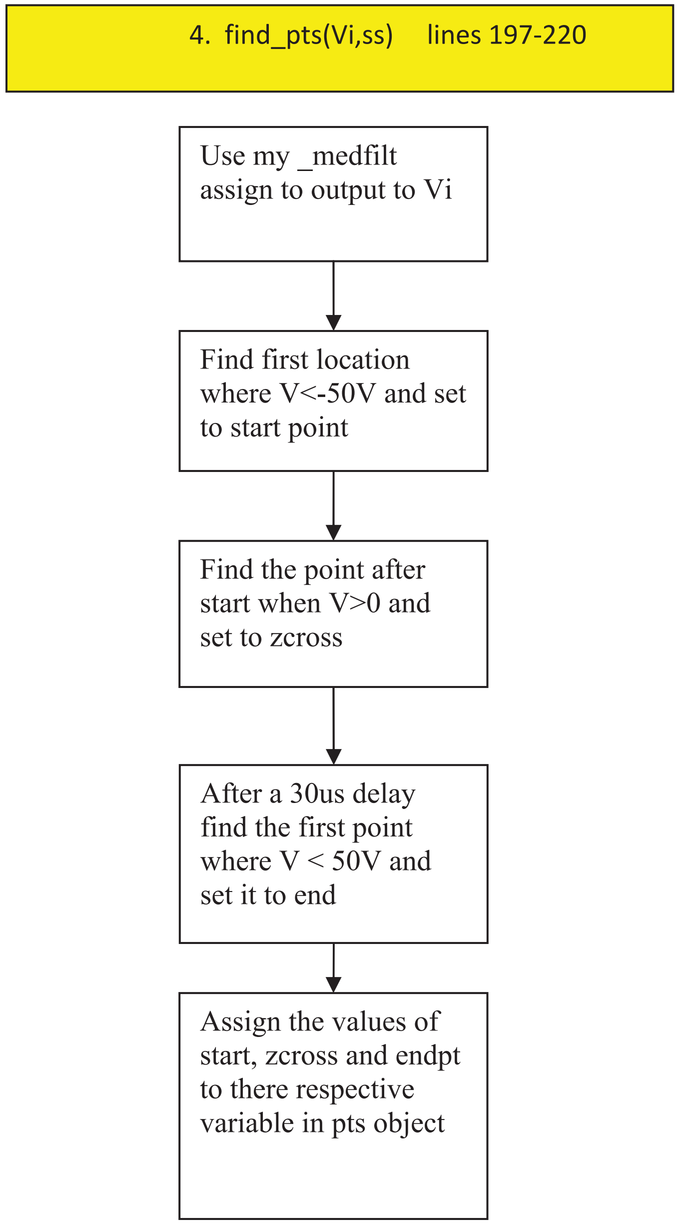


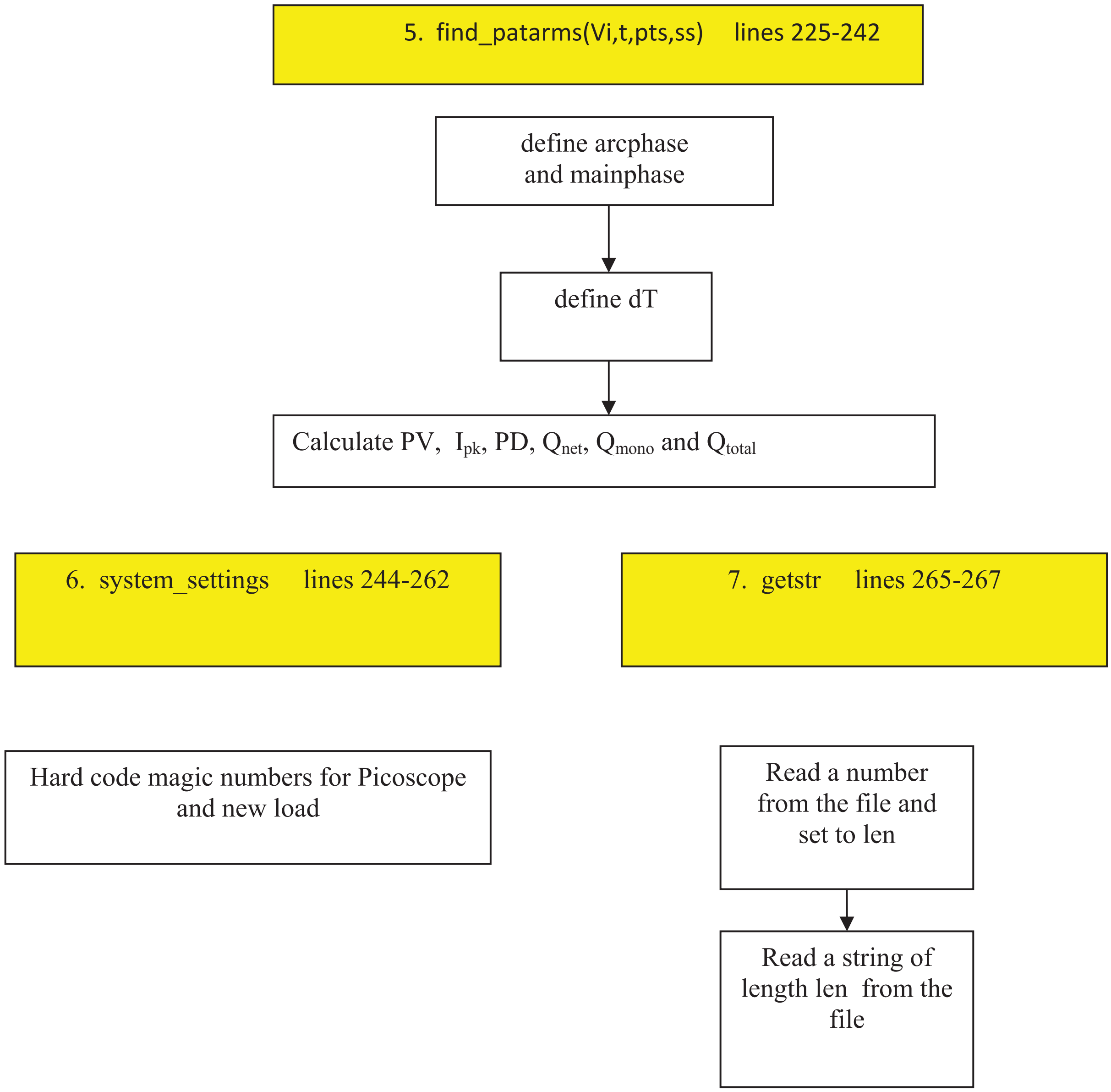
2. print_outputs(params,ss) lines 70-173



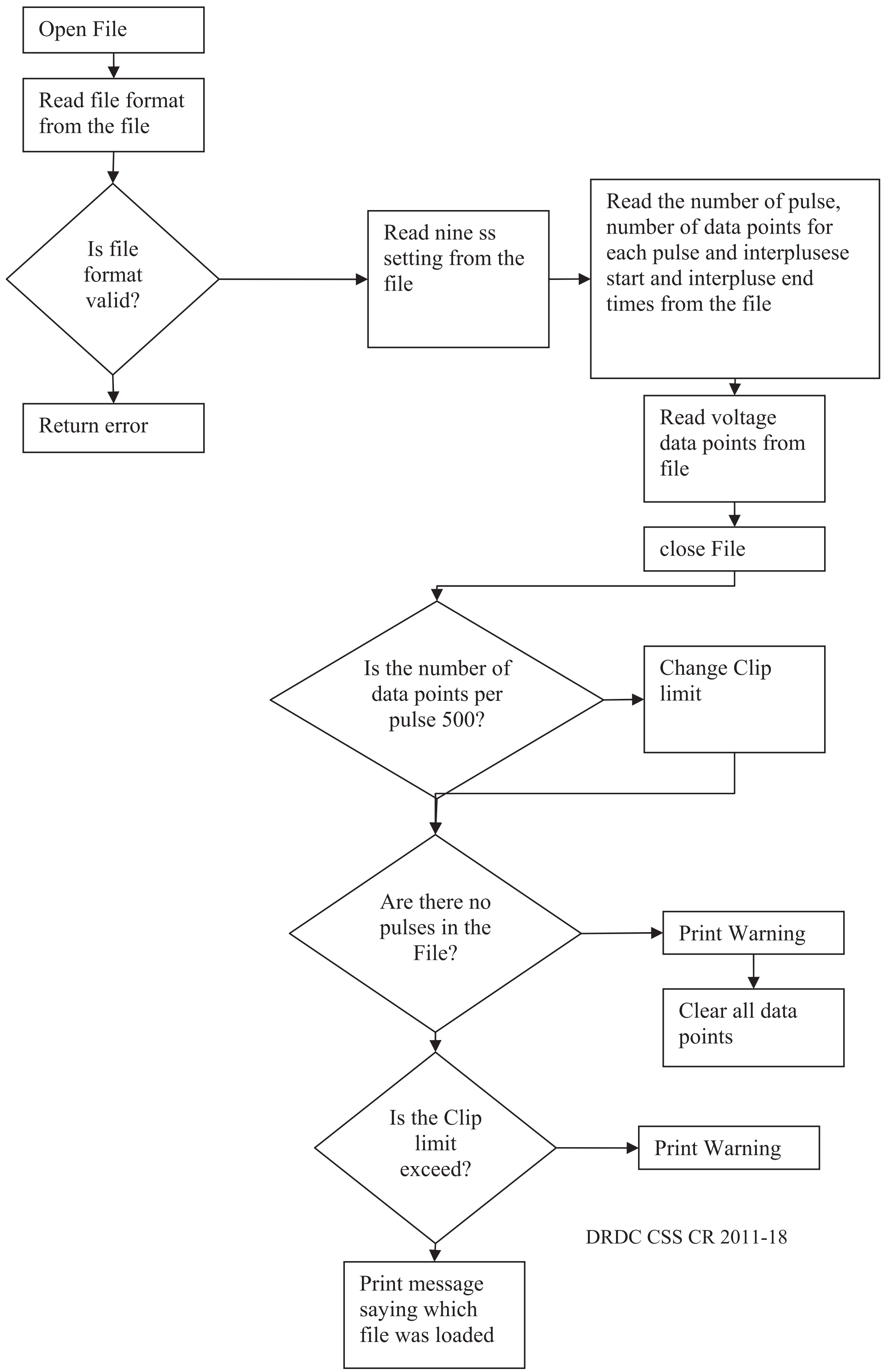
3. outlines lines 177 – 193



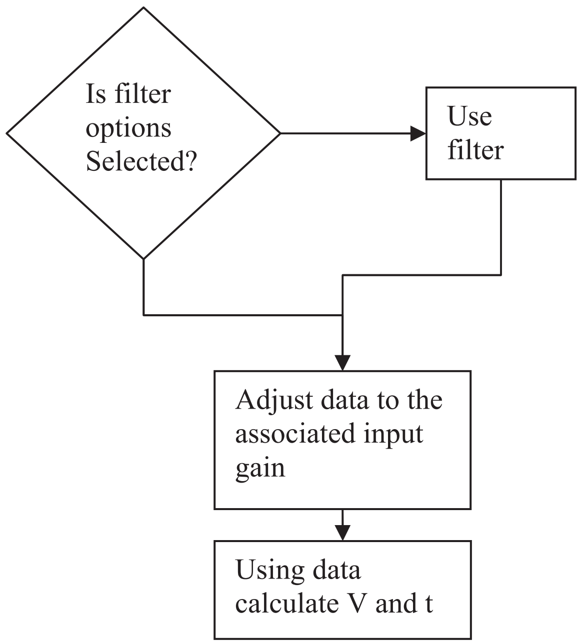

4. find_pts(Vi,ss) lines 197-220
Use my _medfilt assign to output to Vi
Find first location where V<-50V and set to start point
Find the point after start when V>0 and set to zcross
After a 30us delay find the first point where V < 50V and set it to end
Assign the values of start, zcross and endpt to there respective variable in pts object


5. find_patarms(Vi,t,pts,ss) lines 225-242
define arcphase and mainphase
define dT
Calculate PV, I_{pk}, PD, Q_{net}, Q_{mono} and Q_{total}
6. system_settings lines 244-262
Hard code magic numbers for Picoscope and new load
7. getstr lines 265-267
Read a number from the file and set to len
Read a string of length len from the file

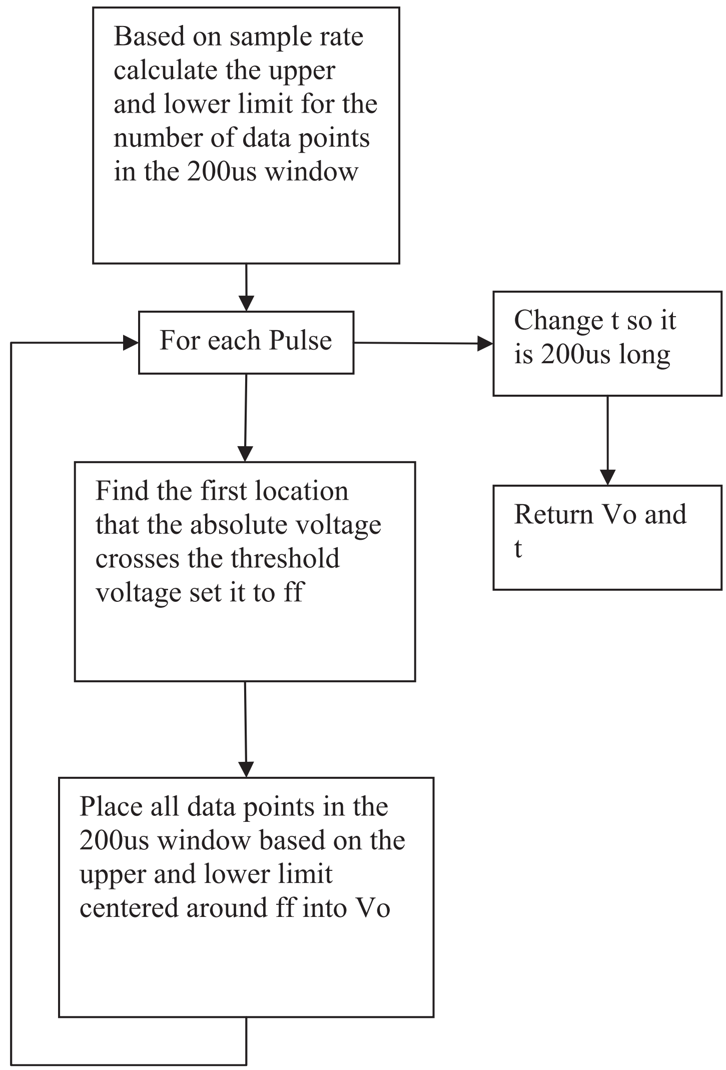
8. read_data (f_name,,ss]   lines 271-305



9. Calibrate_data(data_input) lines 311-321

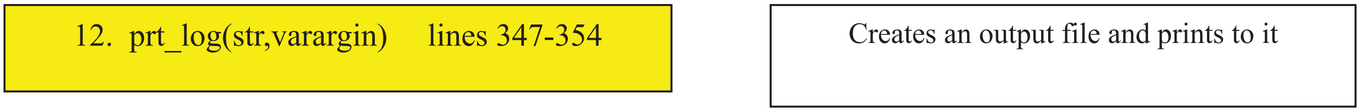


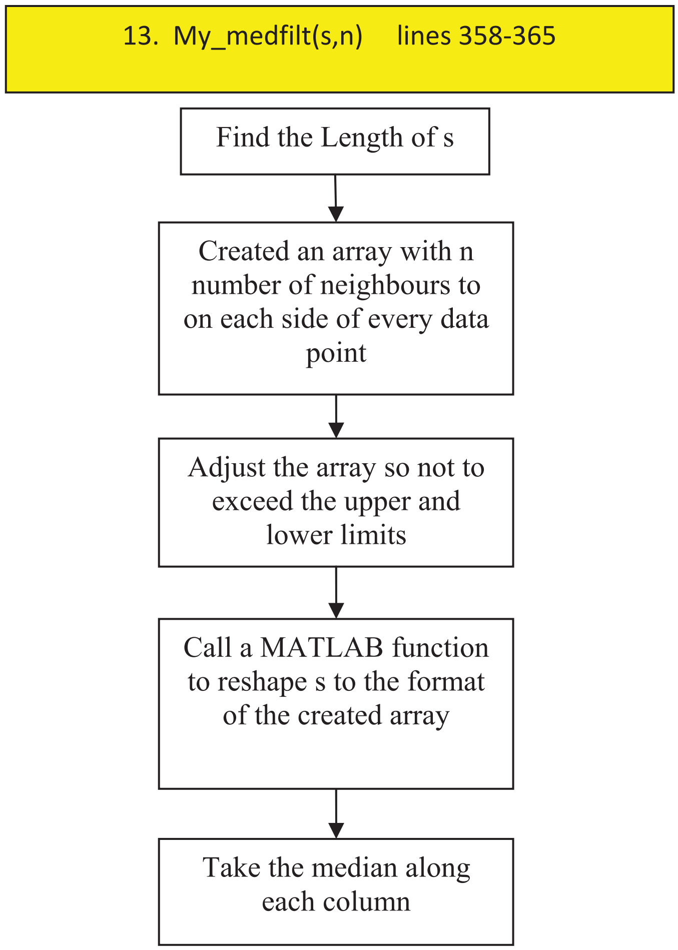
10. retrigger_data(V,t) lines 326-344

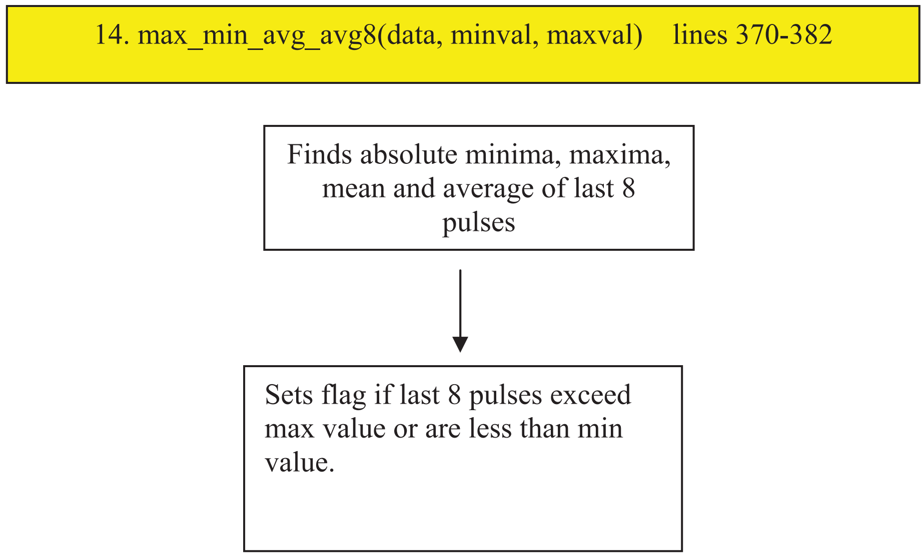


11. retrigger_data(V,t) lines 326-344

Creates an output file and prints to it


12. prt_log(str,varargin) lines 347-354
Creates an output file and prints to it


13. My_medfilt(s,n) lines 358-365
Find the Length of s
Created an array with n number of neighbours to on each side of every data point
Adjust the array so not to exceed the upper and lower limits
Call a MATLAB function to reshape s to the format of the created array
Take the median along each column


14. max_min_avg_avg8(data, minval, maxval) lines 370-382
Finds absolute minima, maxima, mean and average of last 8 pulses
Sets flag if last 8 pulses exceed max value or are less than min value.

List of symbols/abbreviations/acronyms/initialisms

CEW	Conducted Energy Weapon
CEWSI	Conducted Energy Weapon Strategic Initiative
CSS	Center for Security Sciences
DRDC	Defence Research & Development Canada
TI	TASER International

DOCUMENT CONTROL DATA

(Security classification of title, body of abstract and indexing annotation must be entered when the overall document is classified)

1. ORIGINATOR (The name and address of the organization preparing the document. Organizations for whom the document was prepared, e.g. Centre sponsoring a contractor's report, or tasking agency, are entered in section 8.)

   Systems and Computer EngineeringCarleton University 1125 Colonel By Dr. Ottawa, ON K1S 5B6, Canada

2. SECURITY CLASSIFICATION (Overall security classification of the document including special warning terms if applicable.)

   UNCLASSIFIED

3. TITLE (The complete document title as indicated on the title page. Its classification should be indicated by the appropriate abbreviation (S, C or U) in parentheses after the title.)

   Documentation of the Carleton University Conducted Energy Weapons (CEW) Test Analysis Software:

4. AUTHORS (last name, followed by initials – ranks, titles, etc. not to be used)

   Andy Adler; Owen Marsh; David P. Dawson

5. DATE OF PUBLICATION (Month and year of publication of document.)

   October 2011

6a. NO. OF PAGES (Total containing information, including Annexes, Appendices, etc.)

   53

6b. NO. OF REFS (Total cited in document.)

   8

7. DESCRIPTIVE NOTES (The category of the document, e.g. technical report, technical note or memorandum. If appropriate, enter the type of report, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered.)

   Contract Report

8. SPONSORING ACTIVITY (The name of the department project office or laboratory sponsoring the research and development – include address.)

   Project Manager CEWSIDRDC Centre for Security Science222 Nepean StOttawa, ONK1A 0K2

9a. PROJECT OR GRANT NO. (If appropriate, the applicable research and development project or grant number under which the document was written. Please specify whether project or grant.)

   32bj

9b. CONTRACT NO. (If appropriate, the applicable number under which the document was written.)

   W7714-4500833623

10a. ORIGINATOR'S DOCUMENT NUMBER (The official document number by which the document is identified by the originating activity. This number must be unique to this document.)

10b. OTHER DOCUMENT NO(s). (Any other numbers which may be assigned this document either by the originator or by the sponsor.)

   DRDC CSS CR 2011-18

11. DOCUMENT AVAILABILITY (Any limitations on further dissemination of the document, other than those imposed by security classification.)

   Unlimited

12. DOCUMENT ANNOUNCEMENT (Any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11). However, where further distribution (beyond the audience specified in (11) is possible, a wider announcement audience may be selected.))

   Unlimited

13. ABSTRACT (A brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), (R), or (U). It is not necessary to include here abstracts in both official languages unless the text is bilingual.)

This report describes the test analysis software developed at Carleton University (CEW) that was written to support the work on the Conducted Energy Weapons (CEW) Test Procedure, version 1.1 (published 31 July 2010). This report documents the software, its structure, its business rules, and it identifies any data elements required to fully support the published test procedure. This report is designed to support the Conducted Energy Weapons Strategic Initiative (CEWSI), which plans to test a number of CEWs and do analysis on legacy test data with the objective of better understanding and interpreting the performance of the devices, with a goal of developing an updated CEW Test Procedure and providing advice to Canadian Law Enforcement on these devices.

Le présent rapport décrit le logiciel d'analyse d'essai mis au point par l'université Carleton (AI) qui a été élaboré pour appuyer les travaux menés sur les procédures de déroulement des essais sur les armes à impulsions (AI), version 1.1 (publiées le 31 juillet 2010). Le présent rapport décrit le logiciel, sa structure et ses règles administratives et il répertorie également tous les éléments de données nécessaires pour appuyer entièrement les procédures d'essai publiées. Le présent rapport est conçu pour appuyer l'Initiative stratégique sur les armes à impulsions (ISAI), qui prévoit la mise à l'essai d'un certain nombre d'AI et de faire l'analyse des données d'essai des anciens systèmes dans le but de mieux comprendre et de mieux interpréter le comportement de ces dispositifs, avec comme objectif de mettre au point de nouvelles procédures d'essai mises à jour des AI et de pouvoir conseiller les organismes canadiens chargés de l'application de la loi sur ces dispositifs.

14. KEYWORDS, DESCRIPTORS or IDENTIFIERS (Technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible keywords should be selected from a published thesaurus, e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus identified. If it is not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title.)

conducted energy weapons; CEW; Tasers; testing