**S2. Information about the eTape liquid level sensors**

An eTape (Fig. S2.1) is a sensor with a resistive output that varies with the level of fluid in which it is immersed. The resistive output of the sensor is inversely proportional to the height of the water. Low water depths correspond to high output resistance. Conversely, high water depths, correspond to low output resistance.

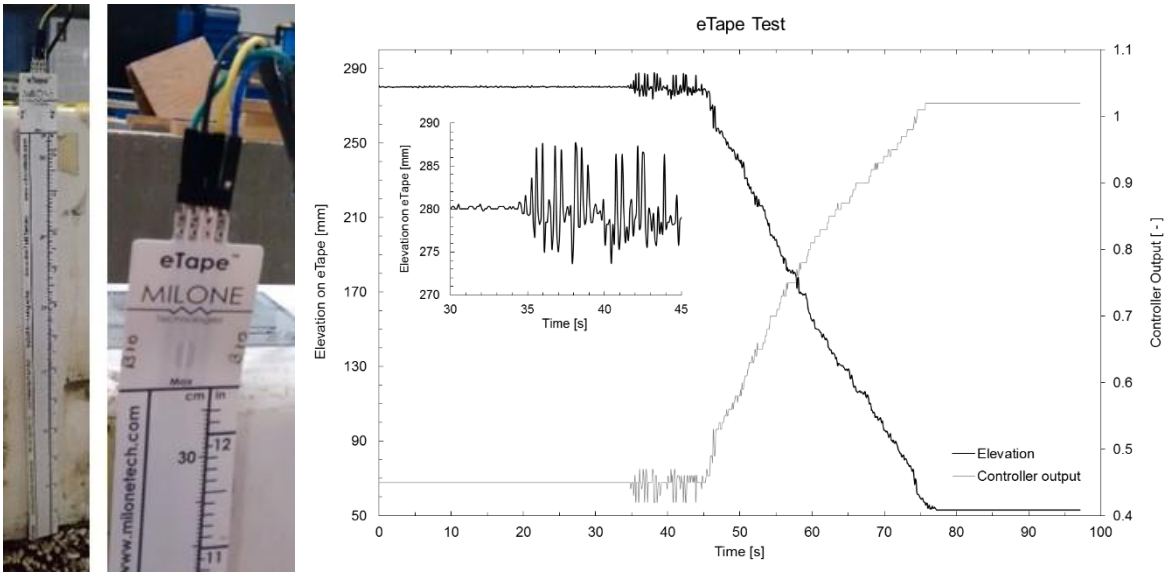
Figs. S1.1a and 1b show an eTape sensor and Fig. S2.1c shows the results of a test conducted with an eTape. The eTape was placed inside a water container. The outlet tube at the bottom was opened at approximately t = 35 s. An air bubble came into the bottle creating the oscillations observed in the curve. A few seconds later, water began to drop down in elevation thus increasing the resistive output from the eTape.

eTapes were installed inside the flume at the locations shown in Fig. 1a in the paper. They were all connected to an Arduino Mega board which was programmed to output to a computer terminal at a frequency of 10 Hz. Fig. S2.2 shows the connections to the Arduino Mega 2560 and Fig. S2.3 shows the wiring diagram. Since this board has 16 analog input pins, it was set up so as to be able to control as many eTapes but in the experiments presented in this study only the four eTapes shown in Figure 1a were connected. All eTapes were connected to the upstream most eTape which served as a reference so as to minimize measurement error due to, for example, water temperature changes.

Conversion of the raw sensor output to water levels required calibration. The calibration was conducted to relate the actual water elevation, as read from the marks printed on the sensor (Fig. S2.1b), to the electric output in the computer terminal. This process was conducted inside the Kinoshita flume. Once the eTapes were installed, the flume was filled to known water elevations in the upstream tank in 1.25 cm (1/2”) intervals. The elevation on each eTape sensor was read and written. The output to the computer terminal was recorded for 300 seconds.

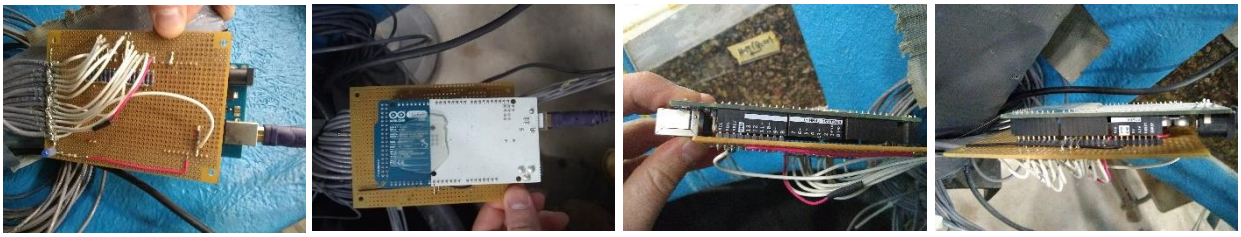
Each 300 second series was loaded into MatLab and the median value was calculated. This median value was associated with the elevation read directly from the sensor. The process was repeated until the range of depths in which the experiments were expected to be conducted was covered. Fig. S2.4 shows the calibration for the eTape located at CS20. A second order polynomial was fit and its coefficients used in the routine to compute instantaneous water surface elevations during the experimental runs.

The Arduino code used is included at the end of this document.




(a) (b) (c)

Figure S2.1 (a) eTape sensor installed in the Kinoshita flume; (b) close-up to the top portion of the eTape sensor; (c) Example of eTape output and actual water elevation during a simple test conducted inside a water container. Curve corresponds to the measurements after the outlet tube was opened.



(a) (b) (c) (d)

Figure S2.2 (a) Top, (b) bottom, (c) right and (d) left sides of Arduino Mega board with all cables connected.

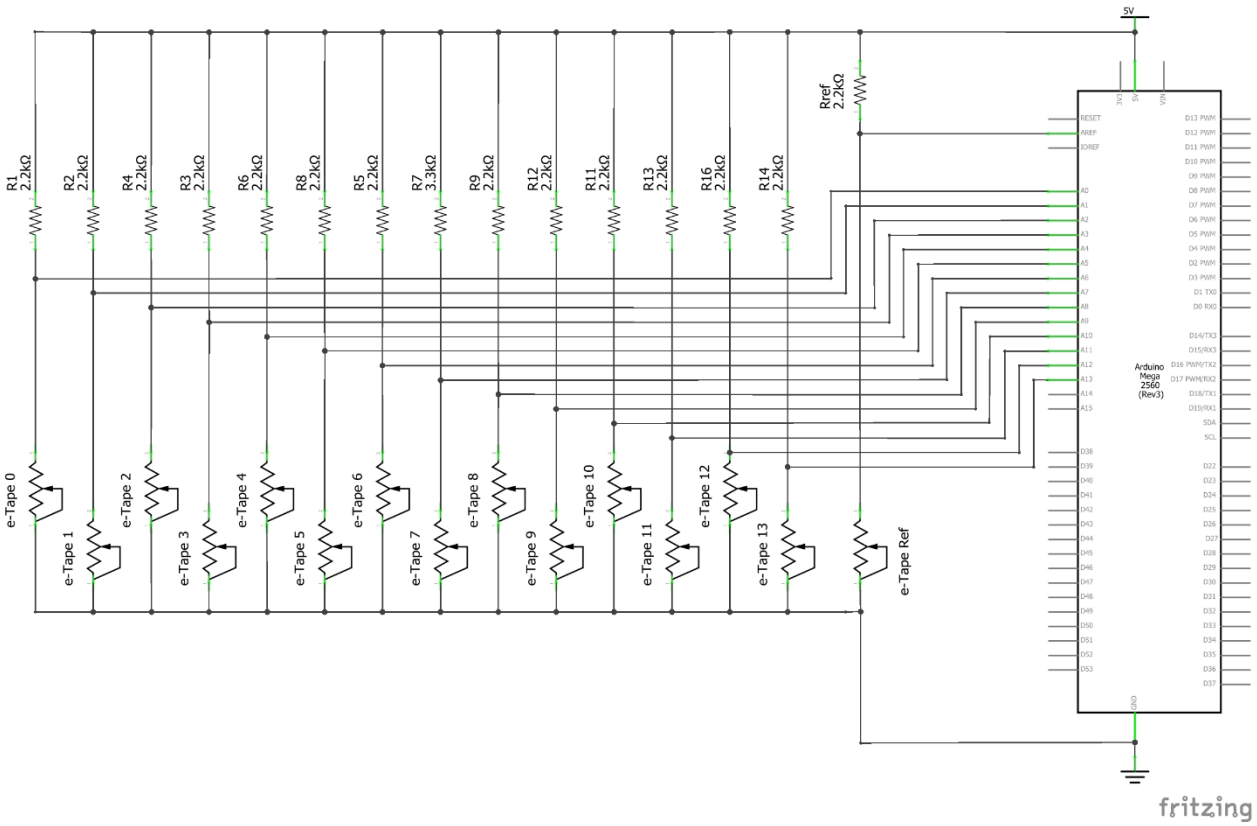



Figure S2.3 Wiring diagram for eTapes in Kinoshita flume. Prepared by Alejandro Vitale.

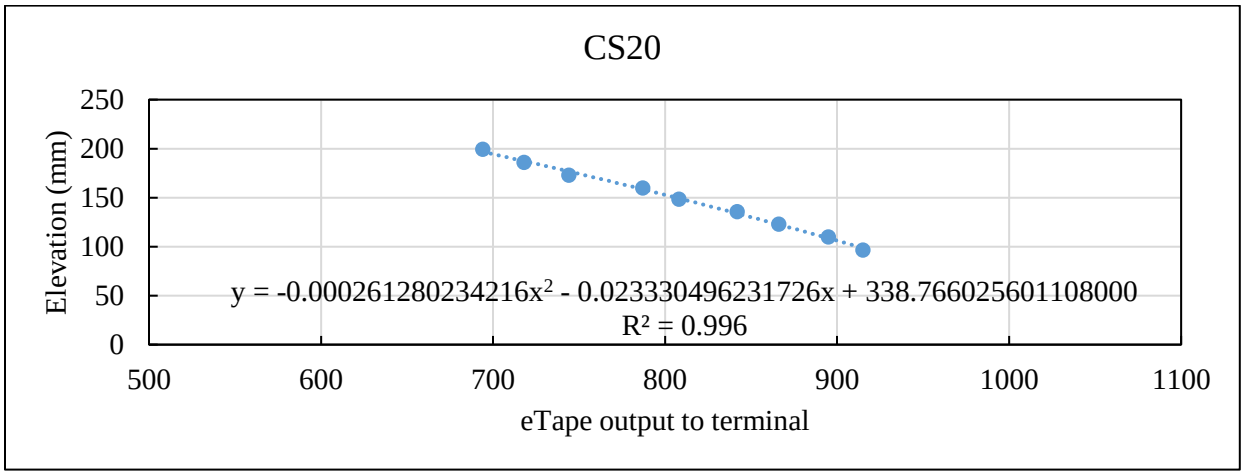



Figure S2.4 Calibration values for eTape located at CS20.

```
/*
 eTape Readings by Dr. Alejandro Vitale and R. Fernández
 Reads analog inputs from the eTape liquid level sensor
*/

// Include
#include <MsTimer2.h>

// Def
#define          NETAPE          16//  number  of  eTapes
#define LED_PIN 13
#define          TM          100// msec sample period

// Global Vars
float                raw_eTape[NETAPE];// Raw eTape data
boolean      blinkState =true;         //  Led  state

//  Setup
void setup()
{
  // Initialize serial communication
  Serial.begin(57600);

  // Conect one eTape ref to Arduino Aref
  analogReference( EXTERNAL );

  // Timer 2 Interrupt
  MsTimer2::set(TM, timerInt);// 100ms period
   MsTimer2::start();

  //
  pinMode(LED_PIN, OUTPUT);
  digitalWrite(LED_PIN,   blinkState);// Led On
```

```
}

// Timer 2 Interrupt function
void timerInt(){
  ad_convert();
 WriteToTerminalInst();
}

// A/D Conversion from eTape
void ad_convert(){
  for (int i=0; i<NETAPE; i++){
     raw_eTape[i] =analogRead(i);
     }
}

// Write To terminal Instantaneous data
void WriteToTerminalInst(){
  //  milliseconds
  float ms  =millis();
  Serial.print( ms / 1000 );
  Serial.print( "\t");

  for (int i=0;i<NETAPE; i++){
    Serial.print( raw_eTape[i] );
    Serial.print( "\t");
   }
  //
  Serial.println();

  // blink LED to indicate activity
  blinkState = !blinkState;
  digitalWrite(LED_PIN, blinkState);
}
```